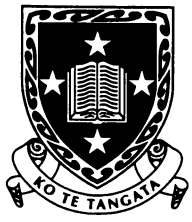

The University of Waikato
Te Whare Wānanga o Waikato

Exploring Usability Evaluation of Localised Software in Malaysia

Alvin Wee Yeo

This thesis is submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy in Computer Science at The University of Waikato.

September 2002

Abstract

This thesis postulates the theory that software usability evaluation should be adapted to the culture in which it is carried out. An experiment has been conducted to explore the effectiveness of usability assessment tools (UATs). These UATs are studied in the context of usability evaluation phase of the global-software development life-cycle. In particular, the study investigates whether imported UATs were appropriate in the usability evaluation of a localised spreadsheet in Malaysia. The experiment reveals that there is some support for the thesis and recommends a further large-scale formal study.

Dedication

This thesis is dedicated to my Mum, and in memory of my Dad.

Thank you for all the sacrifices you have made.

Acknowledgments

First and foremost, I would like to thank Dr. Robert Barbour, my Chief Supervisor for his phenomenal patience, tireless effort, prompt feedback, guidance, encouragement, and advice. No matter how busy he is, he is always free for a quick question or a five-minute meeting which usually lasts longer than five minutes! Without Bob's support and supervision, this thesis would not have seen the light of day. Thank you very much, Bob!

I would also like to thank Keith Hopper, my second supervisor, for providing valuable and insightful comments on my work, advising me on writing the thesis, guiding me through this "mystifying" language called English. Thanks also go to my third supervisor, Professor Mark Apperley, for his feedback on and effort put into earlier versions of this work, and also for the generous funding to attend conferences, both locally and abroad.

This thesis would not have been possible without the scholarship from *Universiti Malaysia Sarawak* (Unimas). Members of the Study Leave Committee, and staff of the Human Resource Division, and the Finance Division at Unimas have been very helpful during the course of my study. I am indebted to Professor Khairuddin Ab Hamid, former Dean of Faculty of Information Technology (FIT) and current Deputy Vice-Chancellor (Development), for his support, understanding, encouragement and patience during the duration of my study. I also appreciate the continuing support of the current FIT Dean, Associate Professor Narayanan Kulathuramaiyer. Thanks also go to FIT's academic, administrative and technical support staff who have been supportive and helpful during my part-time study. In particular, I would like to thank: Dr. Wang Yin Chai, Dr. Rosziati Ibrahim, Dr. Wong Chee Weng, Faaizah Shahbodin, Victor Chee, Halikul Lenando and Sylvester Arnab.

At the University of Waikato, I would also like to thank the Computer Science (CS) Department, Professor Mark Apperley (Chairperson), Associate Professor Geoff Holmes (Acting Chairperson), and Phil Treweek (Senior Tutor) for the Graduate Assistant/tutor appointments. Tutoring was easier with guidance from Phil Treweek, Nilesh Kanji, Tim Elphick and Bronwyn Poki, and fun given the camaraderie among the tutors. The academic and administration staff at the CS

Department have been extremely friendly and supportive. Thanks go to Associate Professor Geoff Holmes, Dr. Eibe Frank, Bill Rogers and Dr. Mark Hall for taking an interest in my progress. I appreciate the help given to me by Marian McPherson, Angela Powers, Pauline Wilson, and Emma Quinlan. Writing the thesis would have been a much more harrowing experience without the help of the technical support staff, in particular, Roger Thomas, Dale Fletcher, and David Baird, who at one time or another, kept my PC and Mac running. Thanks also go to all the graduate students (Anette, Dragan, Stuart Y.), the soccer-playing staff and soccer-playing ex-graduates, who have made graduate-life endurable.

I would like to acknowledge the feedback and assistance of: Dr. Judi McWhirter and Dr. Ray Littler (Statistics Department), Associate Professor Terry Crowley (Linguistics Department), and Dr. Mark Utting (Computer Science Department). I am grateful to Dr. Paula Bourges-Waldegg, Dr. Kumiyo Nakakoji, and Professor Masaaki Kurosu for providing valuable feedback on my work. Thanks also go to Dr. Pia Honold for providing copies of her papers.

I am indebted to Kirsten Thomson and Dr. Sally-Jo Cunningham, for reading and providing valuable feedback on my thesis, and also for the encouragement at the last stage of my write-up. I would also like to thank Dr. Masood Masoodian for his advice, support, and his stress-relieving sense of humour.

My sincere appreciation goes to my housemates, Dr. Sayeeda Bano, Zafar, and Irshad, who accepted me as part of their family. To my email "support group", Beverly LaRock, Bong Chee Thin, Jeremy Chan, Hii Tiong Kong, and Andrea Sim who have cheered me from afar, thanks.

I am fortunate to receive the unconditional support and encouragement given by my in-laws, Uncle Martin and Auntie Margaret, my siblings and my Mum. And finally, to my wife, Wai Ling, for her strength, motivation, and simply being there with me throughout this "roller-coaster ride" called a PhD, *thank you*!

In closing, I'd like to dedicate this Malay *pantun* (quatrain) to all of you:

Pisang emas dibawa belayar	With golden plantain sail away
Masak sebiji di atas peti	Whilst on a chest lies one that's ripe
Hutang emas boleh dibayar	The debts of gold we can repay
Hutang budi dibawa mati	But debts of kindness last throughout life

Sourced from Asma Abdullah (1996)

Contents

List of Figures

List of Tables

Publications

The following publications have been prepared in the course of this research:

1. Yeo, A. (2001). Global-Software Development Lifecycle: An Exploratory Study. In Jacko, J., Sears, A., Beaudouin-Lafon, M. and Jacob, R. (Eds.). *Proceedings of CHI 01: Conference on Human Factors in Computing Systems.* (Seattle, USA, 31 March-5 April). ACM Press. p104-111.
2. Yeo, A. (2000). Usability Evaluation in Malaysia. *Proceedings of 4th Asia Pacific Computer Human Interaction Conference: APCHI 2000.* (Singapore, 27 November-1 December). Elsevier. p275-280.
3. Yeo, A. (1998). Cultural Effects In Usability Assessment. In Karat, C. and Lund, A. (Eds.). *Proceedings of CHI 98: Conference on Human Factors in Computing Systems (Summary) – Doctoral Consortium.* (Los Angeles, April 1998). ACM Press. p74-75.
4. Yeo, A., Barbour, R. H. and Apperley, M. (1998). Cultural Influence in Usability Assessment. In Hanson, M. (Ed.). *Contemporary Ergonomics 1998.* (Cirencester, Gloustershire, UK, April). Taylor and Francis. p274-278.
5. Yeo, A. and Barbour, R. H. (1997). Language Use in Software. *Proceedings of the International Conference of Computing in Education.* (Kuching, Malaysia, December). Association of the Advanced Computing in Education. p862-870.
6. Yeo, A. and Barbour, R. H. (1997). *Localising a Spreadsheet: An Iban Example. Malaysian Journal of Computer Science.* Vol. 10. No. 2. p38-44.
7. Barbour, R. H. and Yeo, A. (1996). Internationalising a Spreadsheet for the Pacific Basin Languages. *New Zealand Journal of Computing.* Vol. 6. No. 1. (December 1996). p14-25.
8. Yeo, A. (1996). Software Internationalisation and Localisation. In Grundy, J. and Apperley, M. (Eds.). *Proceedings of the OZCHI'96: The Sixth Australian Conference on Computer-Human Interaction.* (Hamilton, New Zealand, 24-27 November). IEEE. p65-66.
9. Barbour, R. H. and Yeo, A. (1996). Bilingual/Multi-Lingual Business Software: the Spreadsheet. In Sallis, P. (Ed.). *Proceedings of the 1996 Information Systems Conference of New Zealand.* (Palmerston North, New Zealand, 30-31 October). p63-71.
10. Yeo, A. (1996). Cultural User Interfaces: A Silver Lining in Cultural Diversity. *SIGCHI Bulletin.* Vol. 28. No. 3. p4-7.

Chapter 1 Introduction

1.1 Rationale for the Research

Many developing countries are adopting Information and Communication Technologies (ICTs) in the hope of reaping the benefits that go with them. These nations expect that ICTs will improve their lives and help in solving problems such as poverty, disease and inadequate housing.

There are indeed success stories that have shown that people's lives have been changed for the better through the adoption of ICTs. For example, in the Congo, people who access the Internet via short-wave radio can get information about clean water and health. In India, a series of CyberKiosks link people to district administrators via an Intranet (Rai, 2001). Using these CyberKiosks, the farmers can check the prices of wholesale produce to ensure that they were not short-changed by middlemen. These farmers can also sell land and cattle as well as obtain official documents using these kiosks (Rai, 2001).

Besides improving people's lives, developing countries which adopt ICTs also expect to grow and prosper as richer developed nations have done. Developing countries which do not embrace modern technologies risk continuing deprivation and poverty for their peoples as well as risk being excluded from mainstream economic trends of the world (Davison et al., 2000).

Before these nations can capitalise on the opportunities provided by ICTs, they must possess a computer-literate workforce. Such a workforce can be acquired through training and education. Given that people learn and progress faster when using software in their own language (Griffiths et al., 1994), localised software is highly desirable for the education of the workforce since such software allows people to interact with it in their local language. A computer-literate workforce would be created faster when the people use these customised software. As such, the availability of localised software for developing countries is integral to the achievement of a computer-literate workforce, which would in turn be able to embrace ICTs and gain the potential benefits offered by ICTs.

Furthermore, the availability of localised software could also bring about other benefits. Many people identify strongly with the culture they were brought up in. A particular community may feel personally rejected and second rate if "their language is not accepted into the magic circle of technology" (Griffiths et al., 1994). These communities may never develop the confidence to adopt ICTs, and thus, significant improvements to their livelihood may not materialise (Griffiths et al., 1994).

Also, different people may have different approaches to solutions to the same problems. According to the American Management Association, "heterogeneous work groups create solutions to work and business problems that are more innovative and more effective than those developed by homogeneous groups" (HR Focus, 1993). Thus, by localising software for indigenous groups, software developers have to gather information about these groups, and thus may also be preserving a wider variety of perspectives, potential insights and solutions to the world's problems (Griffiths et al., 1994).

1.2 The Problem

While there is a need for localised software, large software companies usually do not localise software for developing countries as, given their small populations, these countries have a small demand for software. Furthermore, developing nations usually have many small ethnic groups, each with their own language and culture. For example, Papua New Guinea, which has a population of 4.6 million people, has 826 languages. Of these languages, 817 are living languages (Ethnologue, 2001). Although there are languages with large numbers of speakers (for example, Tok Piksin which is spoken by 2 million people), there are many languages that have thousands (some only hundreds) of speakers (Ethnologue, 2001). The likelihood of software being developed for these smaller multiple ethnic groups is even less. Thus, it is usually up to the nationals of the developing countries to provide software for their own people.

Given that the target markets in the world are diverse, developing software to cater for all the different *cultural groups* is a complex process, may incur high costs and may involve significant use of resources. A cultural group is a group whose members have characteristics, such as the ability to speak Swahili,

which distinguishes the group from other groups. In order to ensure successful, effective and efficient provision of software, not only to one cultural group but also to multiple cultural groups, it is necessary for software companies to identify suitable strategies.

1.3 Adapting Software to Multiple Cultural Groups

There are two ways to produce localised software, a one-phase process (retro-fitting) and the current two-phase process (internationalisation and localisation). Language – a component of culture – is used as an example to illustrate the process of internationalisation below.

1.3.1 Retro-fitting

In the past, to obtain software which allows interaction in a different language, the original software was modified directly to encompass the target language interaction. As the original software application was not designed to allow interaction in more than one language, major alterations to the software have to be carried out. As a result of these alterations, different versions of the original software would be created to allow interactions in different languages. Maintenance of the software would be difficult, since each of the language versions had to be updated separately. This one-phase modification process was inefficient, effort and time intensive (O'Donnell, 1994; Madell, Parsons and Abegg, 1994). A better strategy to adapt software from one cultural group to another was put forward. This approach is known as internationalisation and localisation (Hall and Hudson, 1997; O'Donnell, 1994; Uren, Howard and Perinotti, 1993; Taylor, 1992; 1990).

1.3.2 Internationalisation and Localisation

The two-phase approach – internationalisation and localisation – has been promoted as an efficient and effective way to adapt software for multiple cultural groups (Uren, 1998). *Internationalisation* is the process which separates the software into two components, a culture-independent and a culture-dependent component. The culture-independent component, known as the generic core, contains the bulk of the software's functionality and is distinguished from the culture-sensitive elements. To obtain the software for a particular cultural group, the localisation process is conducted. *Localisation* is defined as the process of

providing the culture-dependent components for a target cultural group. In current practice, culture-sensitive elements (comprising dialogue messages, error messages, and menu names) are localised and stored in a message file. (The message file is a manifestation of the utterances with which this thesis is concerned.) There is typically a different message file for each of the different target cultural group.

Thus, before software is provided to multiple target cultural groups, the software is first internationalised. To obtain software that allows users to interact with the software in other languages, the only additional effort required is the localisation phase. There is no modification of the software's generic core. Unlike the one-phase retro-fitting that results in many different versions to cater for the different target groups, there is only one version of the culture-independent application – the internationalised software. Therefore, only one version of the software needs to be maintained and updated. The maintenance of only one version translates into significant savings in both time and money.

1.4 The Problem Context

To develop software for many cultural groups, software developers define separate internationalisation and localisation phases in the design process, in addition to the phases found in a conventional software development methodology. To aid readability of this thesis, a conventional software development process which includes the internationalisation and localisation phases will be referred to as a global-software development process. In this thesis, the global-software development process is the process by which software for multiple cultural-groups is produced. This employs the methodology known as the global-software development life-cycle (global-SDLC), in the production of software for multiple cultural groups. The word "global" in "global-SDLC" indicates that the life-cycle can be applied to develop software for one or many cultural groups in the world. By following the global-SDLC guidelines, software engineers would be able to develop software targeted at numerous cultural groups.

At present, although there exists much information about internationalisation and localisation, little has been reported about global-SDLC. It is expected that the global-SDLC, an invention of developed nations, can be

employed in all cultural contexts to provide software for different cultural groups. However, reported anecdotal evidence indicates that tools employed in usability evaluation of software may not work in cultural contexts outside that of the tools' origin. For example, interviews may not work in Japan. Nakakoji (1994) pointed out that if Japanese users said they liked a system, it is possible that either the test users may be telling the truth or the users were too polite to make negative comments about the system.

Given the anecdotal evidence, the global-SDLC itself may not be effective when employed in cultural contexts outside the global-SDLC's origin, for example, in developing countries. Since software may need to be localised to each target cultural group, the global-SDLC may also have to be adapted to each target group before it can be employed in that target community.

1.5 Thesis Statement

The claim of this thesis is that the global-software development process, as currently defined, has to be adapted in its entirety for the target cultural group before it can be employed in migrating software from a source cultural group.

1.6 Argument of This Thesis

To obtain evidence to support or refute the above claim, an exploratory study will be conducted by applying the global-SDLC to migrate software from a source cultural group to multiple target cultural groups. The experiment will re-engineer an existing spreadsheet (originally targeted at US English speakers) to the following target cultural groups: Bahasa Melayu and Iban speakers. Bahasa Melayu is Malaysia's national language and Iban is a language of a particular Malaysian ethnic group. This spreadsheet will be designed and implemented following the steps of the global-SDLC identified above.

Although the whole global-SDLC is applied to the re-engineering process, the main focus of the research is on the three usability assessment tools (UATs) employed in the usability evaluation phase. The UATs are applied to evaluate the usability of *Hamparan*, a spreadsheet targeted at Bahasa Melayu speakers. The three UATs are logging-augmented think aloud, System Usability Scale (a

questionnaire), and interview. The usability data collected from the three UATs should indicate the success of the software. If the software is successful, it is highly likely that the sub-processes in the global-software development process (such as design, implementation, internationalisation and localisation phases) applied to adapt the software, has also been successful.

With regard to the claim of the thesis, the effectiveness of the UATs employed in the usability evaluation is examined. If the UATs are effective in collecting data and the data collected indicates that the software developed is successful, then processes in the global-software development process are also deemed to be successful. If the global-SDLC is successful, the global-SDLC is deemed successful without any adaptation. Such a finding would refute the claim of this thesis.

However, if the UATs are not effective in the usability evaluation, the usability evaluation phase will be deemed ineffective. A re-examination of the data collected will have to be carried out to identify reasons for the UATs' ineffectiveness. It may be that components of the usability evaluation phase may require adaptation. If the process employed in the usability evaluation phase is adapted and the usability data collected from the usability evaluation confirms the success of the software, then the global-SDLC, with an adapted usability evaluation phase, is also deemed to be successful. Such a finding would partially support the thesis's claim; further work to look at the remaining global-SDLC phases would then be required.

1.7 Thesis Structure

There are ten chapters in this thesis (see Figure 1.1). This chapter introduces the context of the research, that is, the motivation for the research and the claim of the thesis. Chapter 2 reports on a review of the literature associated with the global-SDLC, and its component processes. Chapter 3 introduces the key question of the thesis and suggests an answer. The answer involves obtaining information from the usability evaluation, which indicates the success of the global-SDLC. Chapter 4 reports on the steps employed to internationalise and localise a spreadsheet. Chapter 4 also details the steps taken to evaluate the effectiveness of three UATs in the usability evaluation of the localised

spreadsheet, *Hamparan*. Results of the usability evaluation are presented in Chapter 5. An analysis of the results is reported in Chapter 6. A discussion of the analysis, detailed in Chapter 7, indicates that the cultural background of the participants may explain the anomaly found in the data. To confirm this suggestion made in Chapter 7, the usability data was re-examined. The results and discussion of this re-examination are detailed in Chapter 8. Chapter 8 also describes the implications of the findings on conducting usability evaluation in Malaysia. Chapter 9 presents research findings in relation to the global-SDLC. Finally, Chapter 10 concludes the thesis and suggests further work.

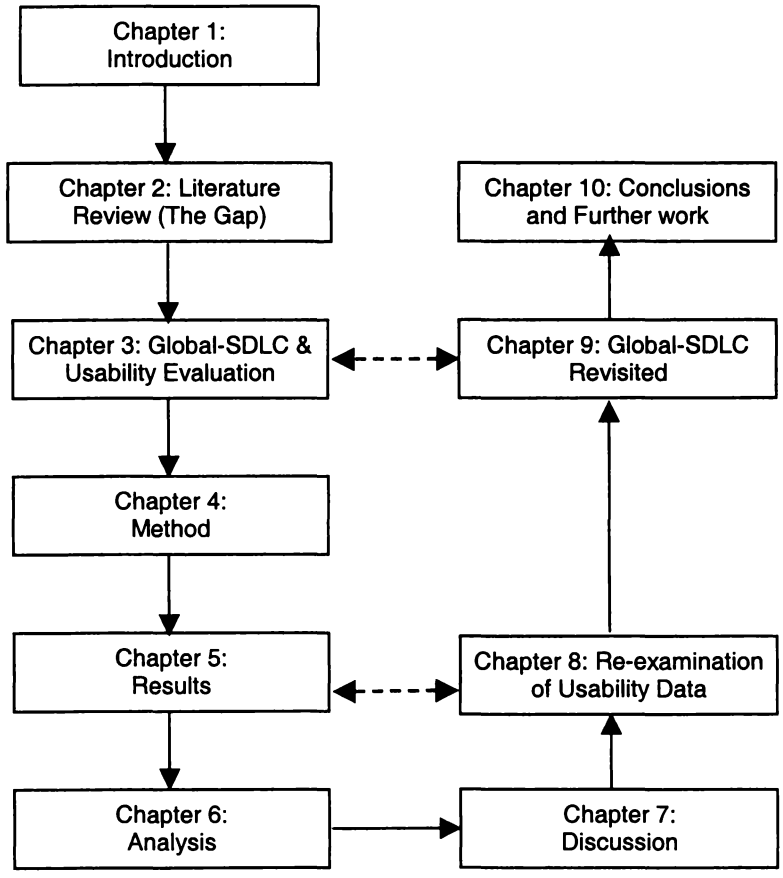

Figure 1.1: Structure of the Thesis

Chapter 2 Literature Review

In Chapter 1, a need for improved approaches to develop software for multiple cultural groups was identified. In this chapter, published works in this field of research and practice are surveyed by reviewing relevant publications and presenting them in a chronological order. This review can be presented in a chronological order or ordered thematically. A thematic presentation makes it easier for the reader to follow as the comparison of the works is carried out by the author. Note however, such a presentation means that the author is imposing his (culturally-biased) view on the reader. Using a chronological presentation avoids such an imposition; readers are allowed to compare the literature in a way that is culturally acceptable to them. Furthermore, publications reviewed in a chronological order shows the historical development of experts' thinking.

To provide a context for this literature review, a definition of "culture" is supplied in Section 2.1. General Systems Theory, the theory on which systems engineering and consequently, software engineering were based on, is outlined in Section 2.2. Sections 2.3 and Section 2.4 describe connections between current software engineering models and the provision of software for one and then multiple cultural groups, respectively. Section 2.5 provides a review of existing literature pertaining to approaches and related activities for providing software to multiple cultural groups. A summary of the literature reviewed is given in Section 2.6. This chapter ends with Section 2.7, in which the research focus of this thesis in relation to the work of others is outlined.

2.1 Definition of Culture

In this thesis, the term "culture" is defined as learnt behaviour comprising thoughts, feelings and actions (Hoft, 1996). This learnt behaviour distinguishes the members of one *group* of people from another (Hofstede, 1994). This group, henceforth known as a *cultural group*, consists of people who share the same culture, that is, they think, feel and act in a similar manner. Cultural groups are defined by factors such as nationality, geographical location, or ethnic groups. Cultural groups can also be defined by occupations, the organisations people work in, or the expertise/roles of people in work settings. As such, a person may belong

to numerous cultural groups. For example, Ariunaa is a Mongolian who works as an architect in Mongolia. In this case, she is a member of at least two cultural groups, that of Mongolians, and of architects. She would most likely know and understand the rituals and norms of Mongolians. However, she would also possess knowledge about architecture; knowledge she shares with architects all over the world.

Irrespective of how a cultural group is defined, the cultural differences between groups can be categorised into symbols, heroes, rituals, and values. These categories are also called manifestations of culture, see Table 2.1 (Hofstede, 1994).

These categories can be considered as layers of culture, much like, skins of an onion, see Figure 2.1. The symbols layer is the most superficial. "Peeling" the symbols layer will reveal, heroes and so on (Hofstede, 1994). The outer layer would be the most visible or overt, and values layer the most hidden. All symbols, heroes, and rituals layers are included under the term "practices". As such, outsiders can see these practices. However, the cultural meanings of these practices are invisible to the outsiders (Hofstede, 1994).

Symbols include words, gestures, pictures and objects that carry a particular meaning, which is only recognised by those who share the same culture.
Heroes are persons, alive or dead, real or imaginary, who possess characteristics which are highly prized in a culture, and who serve as models for behaviour.
Rituals are collective activities, technically superfluous in reaching desired ends, but which, within a culture, are considered as socially essential, for example, ways of greetings.
Values are broad tendencies to prefer certain states of affairs over others. Values have a plus and a minus side. For example, evil vs. good, dirty vs. clean, ugly vs. beautiful. Norms are the standards for values that exist within a group or category of people ... norms pertain to what is ethically right.

Table 2.1: Hofstede's (1994) definitions of cultural manifestations

In relating the cultural group to the cultural manifestations, every member of a cultural group would possess recognisably similar attitudes and behaviours, as well as think and act in recognisably similar ways given the same situations. Furthermore, members of a cultural group are likely to perceive an artefact as having the same significance. However, members of a different culture may not

perceive the artefact as having the same significance. Using an example from the symbols layer, an Algerian would understand a sentence in Arabic, as would most other Algerians. However, a Thai, who most probably has not learnt Arabic, would not understand the same Arabic sentence. The Algerian has learnt the meanings assigned to the Arabic words, and he would know what the combination of Arabic characters mean. The Thai would not understand or recognise the Arabic sentence, as most Thais have never learnt Arabic. Thus, the perception of members of one cultural group literate in Arabic, would not be the same as compared to the perception of members of another cultural group, who do not know Arabic.

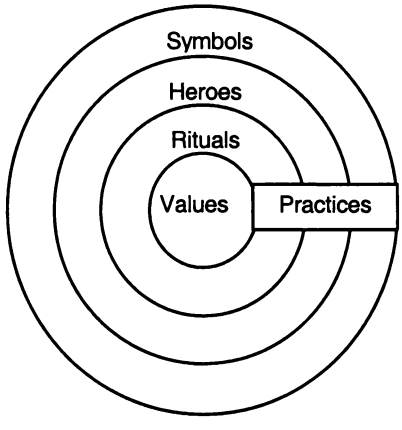

Figure 2.1: Layers of manifestations of culture (sourced from Hofstede (1994))

While the above example would seem to show that different cultures are associated with nations, cultural groups can be "categorised" differently as illustrated by an example in Bødker and Pedersen (1991). Bødker and Pedersen describe a pump-valve which sits on a pedestal behind glass – like a precious ornament – in the lobby of a company. The valve was the first pump-valve from the company's new supplier. To the members of the workplace, the valve symbolised "autonomy and independence".

A newcomer to the company would probably assume that the pump-valve was important because it was the first component manufactured in the company. Just like the Mongolian architect example, this example also illustrates that cultural groups need not necessarily be associated with nations.

In this section, culture and cultural groups were defined to provide a context for the literature review later in this chapter. Before the literature review

is presented, some background information on systems engineering is provided in the next section.

2.2 Systems Engineering

Given that the software development process originated from systems engineering, which in turn was based on systems theory, it is appropriate at this point to define systems and outline relevant General Systems Theory.

2.2.1 General Systems Theory

General Systems Theory was proposed in the 1940s by Ludwig von Bertalanffy, a biologist (Heylighen, Joslyn, and Tuchin, 1999; Heylighen, 1998). Von Bertalanffy (1968) distinguishes two types of systems, a closed and open system. He considers a closed system as one that is isolated from its environment. These systems do not interact with the outside world (Heylighen, 1998). Von Bertalanffy (1968) gives the example in physical chemistry whereby chemical equilibrium is eventually established within a closed vessel. However, Von Bertalanffy (1968) states that systems "by their very nature and definition are not open systems. ... Every living organism is essentially an open system" (Von Bertalanffy, 1968). Organisms cannot survive without continuously interacting – for instance, exchanging matter and energy – with their environment (Heylighen, 1998). Henceforth, any systems referred to in this thesis are open systems, unless otherwise indicated.

Systems theories propose that a system is an arbitrary assembly or set of related components (Van Gigch, 1974). These components can be concepts, objects or subjects/people, that is, an aggregation of living and/or non-living entities (Van Gigch, 1974). Systems may be made up of other systems, which are known as sub-systems. The total/whole system refers to the systems, which comprise other systems (Van Gigch, 1974). A system exists within a larger world, that is, an environment. The theory focuses not only on the whole system, its sub-systems, but also, the assumed arrangement of, and relations between the sub-systems (Heylighen and Josyln, 1992).

Valacich, George, and Hoffer (2001) describe a system as having nine characteristics. These characteristics are components, interrelated components, boundary, reason for existing/"raison d'être", environment, interfaces, input,

output and constraints. The following description and illustration of systems are based on the work by Valacich, George, and Hoffer (2001).

A system is considered to be a *component*, but a system in turn may be viewed as being made up of other components/sub-systems, which are systems in their own right, see Figure 2.2. These components are *interrelated.* These inter-relationships bind the system together to achieve interactions. All the components combine towards achieving a common *reason for existing.* Anything that enters a system is known as *input.* The input is then "processed or transformed" by the *system.* A result of the process may be *output*, which leaves the system and returns to the *environment.* In this way, the environment interacts with the system.

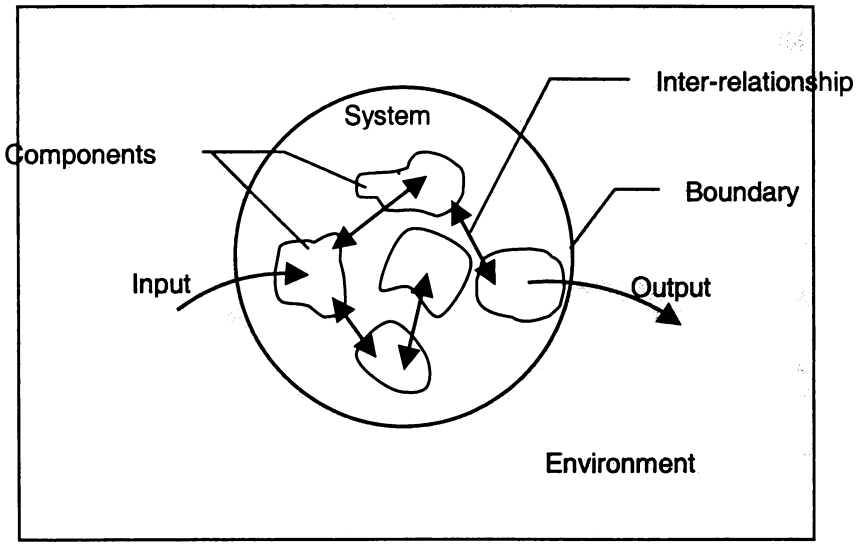

Figure 2.2: Systems adopted from Heylighen (1998) & Valacich et al. (2001)

Using the university as an example of an open system, the university has a reason for existing, which is to teach and conduct research. The university is made up of other components, such as faculties, the human resource division, and the finance division. Each of the faculties and divisions can also be viewed as a system. All the sub-systems of the whole system that is the university are interrelated and support the teaching and the research processes. For example, the human resource division recruits lecturers who work in the faculties. The finance division pays the salaries of these lecturers. Thus, input could be people entering the university/system becoming students who are taught by the lecturers. The students then return to the environment as graduates/output. The environment can be viewed as being made up of numerous other systems. For instance, the environment, itself, could comprise the community, the government, and business

organizations. Thus, the community or businesses could employ the graduates, which are output being returned to the environment and becoming input to other systems.

A *boundary* is defined for the border between the system and environment. This arbitrary boundary is important as it clearly identifies the limits or scope of the system that is being looked at, and separates the system from other systems. With the presence of a boundary, it is possible to distinguish between the university and the systems it is interacting with, that is, its environment. *Interfaces* are where the systems meet the boundary. The *constraints* on the system refer to the limits of the system. For example, a university has limited funding from the government, and may consequently be able to accept only a limited number of students.

By looking at just the system as a whole, one does not need to be aware of the parts that make up the system (Heylighen, 1998). Thus, systems theory provides a way of reducing the complexity and enables focus to be placed on the areas that are of particular interest. For example, if the university is to be made more efficient, only the systems within the boundaries need to be in focus. The systems outside the boundary, such as the community, need not be considered.

In sum, systems theory is "the trans-disciplinary study of the abstract organisation of phenomena, independent of their substance, type, or spatial or temporal scale of existence. It investigates both the principles common to all complex entities, and the usually-mathematical models, which can be used to describe them." (Heylighen and Josyln, 1992). As described, the General System Theory can be applied to any discipline, and the physical, non-physical, social, technical or political systems defined with it. Examples of its use include in the physical sciences (physics and chemistry); life sciences (biology, botany); behavioural sciences (anthropology, psychology); and social sciences (education, applied behavioural science) (Van Gigch, 1974).

According to Jackson (1991), while General Systems Theorists were improving ways of gaining an understanding of social systems, another group of systems thinkers were employing Systems Theory in a much more applied manner. Systems Theory was used to solve real-world problem situations. These systems thinkers were referred to as *hard* systems thinkers and included systems

engineers (Checkland, 1981, p.125). Systems engineering developed from the more traditional engineering disciplines. Jackson (1991) defined systems engineering as "The science of designing complex systems in their totality to ensure that the component sub-systems making up the system are designed, fitted together, checked and operated in the most efficient way." Systems engineering includes the following phases: problem definition, choosing objectives, systems synthesis, systems analysis, systems selection, systems development, and current engineering (Jackson, 1991). These phases are similar to the phases of traditional software engineering approaches. Jackson (1991) points out that although Hall's (1962) systems definition concerned physical entities, there were other applications of systems engineering. One of these applications is software engineering.

2.2.2 Software Engineering

In the late 1960s, large software projects were "consistently late, over budget and full of defects" (Shapiro, 1997, p.20). In particular, only a small percentage of software worked on delivery, the majority of software systems were either, never successfully used, had to be reworked or abandoned, or were never delivered (Mynatt, 1990). There was a need for a disciplined approach to the development of large and complex software systems.

Shapiro (1997) reported recommendations from NATO's (North Atlantic Treaty Organisation) 1968 and 1969 conferences, of "the need for software manufacture to be based on the types of theoretical foundations and practical disciplines that are traditional in the established branches of engineering". By applying sound engineering principles to software development, it was anticipated that good quality software would result (Mynatt, 1990). Engineering and software engineering share their origins from the application of General Systems Theory.

The goal of a software engineer is to produce high quality software that meets users' requirements, is on time and within budget. Developing high quality software is not easy and must be carefully planned; worked on at every stage of the software life-cycle (Mynatt, 1990).

There are three stages in the software life-cycle – development, maintenance and retirement (refer Figure 2.3). In the development stage, software

is developed and tested; in the maintenance and operation stage, software is installed and used; and finally in retirement, software is withdrawn from use.

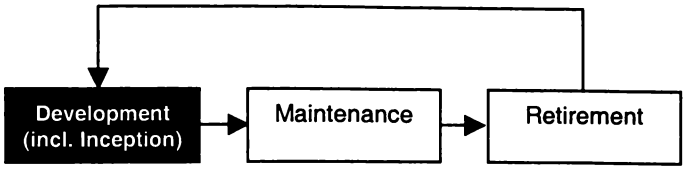

Figure 2.3: Software life-cycle (as compared to software development life-cycle)

Within the development stage, there are a number of activities, such as design and coding, that need to be completed. From 1970 onwards software development activities were described within a framework (Macro, 1990) which provided a better planned and managed way of developing it. The standard ISO/IEC 12207: 1995/Amd. 1: 2002 Information Technology – Software Life Cycle Processes describes the major component processes of software development. It also details tasks and activities in the acquisition, supply, development, operation and maintenance of software. The software development process employs a software development life-cycle in the production of software.

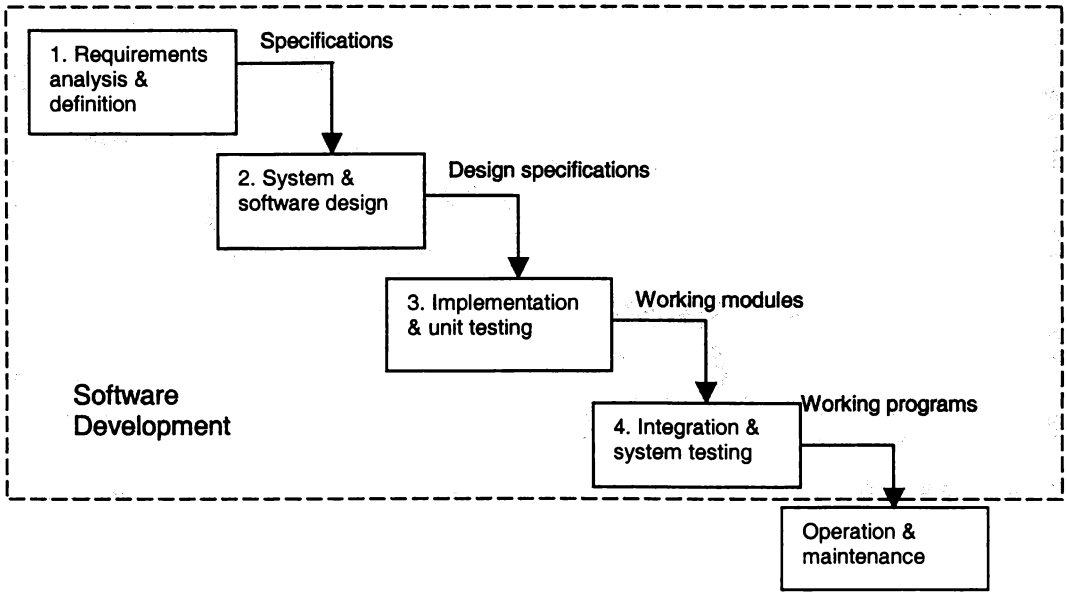

Figure 2.4: "Waterfall" model (sourced from Van Vliet (1993))

There are many software development life-cycle (SDLC) models. Pressman (1997) grouped these life-cycles into a number of categories. These categories include sequential life-cycle models (for example, the waterfall model, as depicted in Figure 2.4), prototyping models, formal methods models,

evolutionary software, process models (such as the incremental model), and spiral models.

One of the earliest SDLCs used was the waterfall model. The basis of this life-cycle consists of four phases of software development: requirements analysis, definition, integration and system testing, see Figure 2.4. Most of the existing models contain all these four phases. What differs between the models is the sequence and frequency of the various phases and the emphasis of certain phases over other phases. Different models are used depending on the context in which the software will be used, as well as the type of software being developed. Details of specific SDLCs can be obtained from software engineering textbooks such as Pressman (1997), Sommerville (1996), Conger (1994), and Van Vliet (1993).

The next sections will describe the differences of developing software for one cultural group, and developing software for different cultural groups.

2.3 Developing Software for A Specific Cultural Group

As defined in Section 2.1, a cultural group is a group of people who think, act and feel similarly. Members of a cultural group also share similar concepts, values and assumptions about life (Jandt, 2001) and work (Hofstede, 1994). Software developed for a particular group of people can be said to be software developed for a specific *cultural group*. For instance, accountants are seen as a cultural group who share the same domain of knowledge that is necessary to talk about their discipline. For the accountants, the domain of knowledge may include similar set of symbols and meanings, terms such as profit and loss accounts, and general ledger entries. Since all SDLCs are employed to build software for a specific group of people, it is argued that all SDLCs are applied to develop software for a specific cultural group. Software developers can apply SDLCs to build software not just for one cultural group, but also for many cultural groups. Details of approaches to provide software for more than one cultural group are provided in the next section.

2.4 Developing Software for Multiple Cultural Groups

There are two ways to develop software for multiple cultural groups: a one-phase and a two-phase approach. In the one-phase process, software

developers customise the software from one cultural group to one or more other cultural groups (see Figure 2.5a). For example, software developers can adapt an English spreadsheet to Spanish, and then they can adapt that English spreadsheet to Portuguese. This process is called retro-fitting and is detailed in the next section, Section 2.4.1. The two-phase approach is elaborated in Section 2.4.2. In the first phase of the two-phase approach, an internationalised software package is developed. This internationalised software can be a result of the internationalisation of existing software (see Figure 2.5b) or can be developed ab initio (see Figure 2.5c). In the second phase, the software is then localised to one or more cultural groups (see Figure 2.5b and Figure 2.5c).

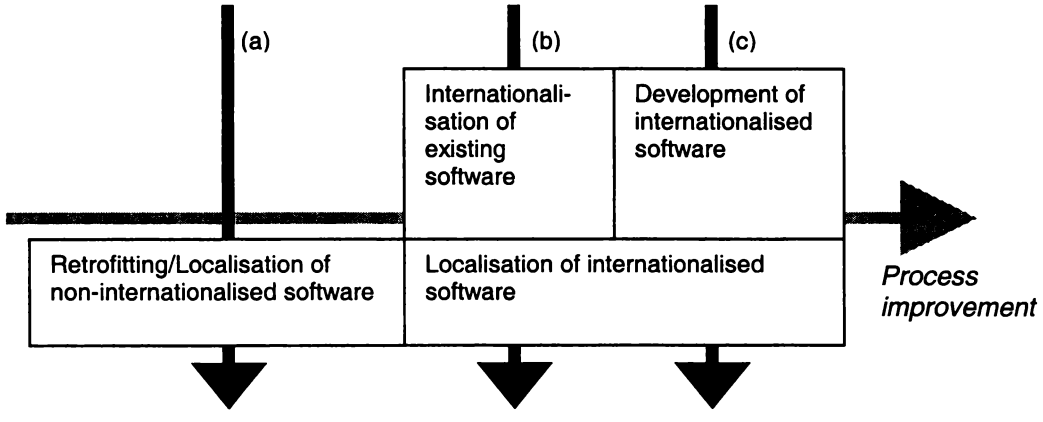

Figure 2.5: Alternatives for l10n process (sourced from Hall and Hudson (1997))

2.4.1 Retro-fitting – the Conventional Approach

To illustrate this approach, the cultural groups used in the following example are distinguished by languages spoken by the groups' members. As many of the major software companies are in the United States, most of the software available in the market today allows users to interact with software in US English. The users of this software are seen as belonging to one cultural group, based on shared language. When a particular software program is required in another language, the software has to be adapted to accommodate the new language. This modification usually means a direct translation of the literals in the original software into the new language. If this modification is not feasible, the software may have to be developed from the beginning to ensure users can interact with the software in the target language. The software is then redesigned and recompiled to obtain the new version in the new target language. If yet another language is required for that software, then the source code must again be altered and recompiled to obtain the new version of the software, see Figure 2.5a.

This strategy of retro-fitting is expensive and time consuming (Madell, Parsons and Abegg, 1994; O'Donnell, 1994). The translated version, which allows interaction in a different language, will always be technologically equivalent to or behind the original version. This lag occurs because a new version of the original software may have been produced by the time the adapted version was completed. Furthermore, as there are many different language versions of the software, maintenance and update of every version is difficult since every version has to be maintained and updated separately (Madell, Parsons and Abegg, 1994).

Given the inefficiency and ineffectiveness of this technique, a new approach was introduced. This approach comprised two phases, an internationalisation phase and a localisation phase.

2.4.2 The Current Approach

The two-stage process, internationalisation and localisation[1], is at present the recommended method for developing software for users all over the world (Honkela et al., 1997; Kano, 1995; Madell, Parsons and Abegg, 1994; Russo and Boor, 1993; Taylor, 1992). *Internationalisation* is defined as the process in which culture-sensitive elements are isolated from the program resulting in two components, a generic core, and a culture-sensitive/dependent component. Culture-sensitive components here refer to cultural elements that are associated with the multiple cultural groups which are the target software users. The culture-independent *generic core* is the program that contains the bulk of the functional source code. This generic core is developed in such a way that it can handle the required cultural elements of the various target cultural groups. Figure 2.6 illustrates a simplified implementation of an internationalised multi-lingual application (Madell, Parsons and Abegg, 1994); the cultural elements are *messages,* literals from the interface components such as error messages, dialogue messages, prompts, and menu names. These messages are stored in *message files* (utterances) that are external to the program. There is typically a different message file for each language. On the other hand, the generic core is common to all language versions of the software. The *language tables* contain all language-

[1] Internationalisation and localisation are sometimes known as i18n and l10n respectively. The figures, 18 and 10, are the number of characters between the first and last characters of the words 'internationalisation' and 'localisation' respectively.

specific processing information and conventions unique to a particular cultural group, for instance, how characters are sorted and how output is formatted.

The Language-Independent Program/generic core in Figure 2.6 is independent of any culture-specific elements such as language. At run time, depending on the language selected by the developer or user, the program will run according to the language table's settings for the specified language. The program will also retrieve the messages from the corresponding language's message file.

To provide different cultural elements for different cultural groups, a process known as localisation is employed. *Localisation* is defined as the process of providing the cultural elements of a particular target cultural group. In current practice, this localisation normally involves translation of the text messages, providing the data display formats (such as date, time, currency and number formats) of the target cultural group as well as other culture-dependent elements such as icons.

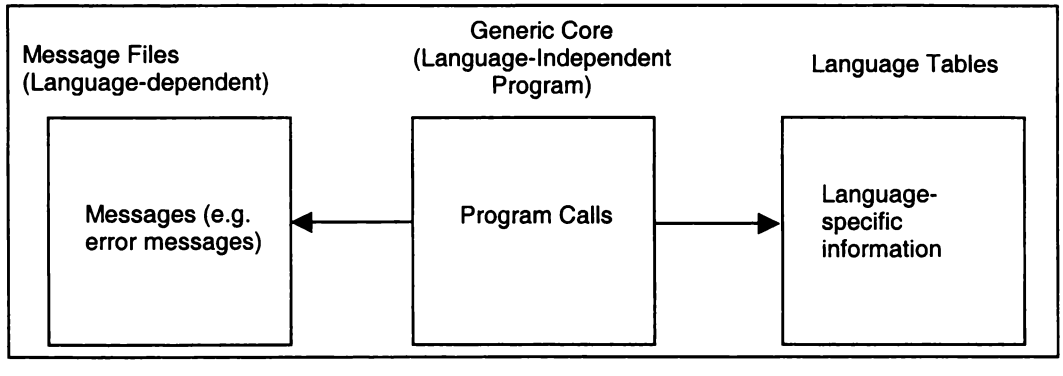

Figure 2.6: Internationalised Model (sourced from Madell et al. (1994))

Thus, to accommodate another language, only localisation takes place. The messages in the original message file are translated to that new language, and placed in a new message file. Language tables are also generated to take into account any unique language conventions of the new language. If software for another language is required, only the localisation process needs to take place; there is no modification of the generic core of the software.

It must be noted that in the current practice of internationalisation and localisation, only issues pertaining to language (text messages) and icons are addressed. Languages and icons are only manifestations of the culture at the symbols layer (as shown in Figure 2.1). These items however make up only a

very small part of the set of cultural elements that should be addressed. Cultural aspects from the deeper layers, which include socio-cultural issues, have largely been neglected.

2.4.3 Advantages of Internationalisation-Localisation over Retro-fitting

The following comparisons, between the internationalisation and localisation approach, and retro-fitting, are adapted from Madell, Parsons and Abegg (1994). These comparisons serve to show the many advantages of this two-phase approach.

Firstly, the non-internationalised software, the US program, can accommodate one character set only, whereas the internationalised and localised software supports a variety of character sets. In addition, the US program can only manipulate and display data according to the rules of US English, whereas the internationalised program can manipulate and display data according to the rules of the user's cultural conventions. Similarly, data display formats are restricted to US English, unlike the localised version, which is modifiable to the user's culture-sensitive requirements.

As the cultural/language component is hard coded in the US English program, each time a new cultural group is targeted, the source code must be altered and recompiled to accommodate the needs of the new target cultural group. On the other hand, to localise the internationalised version, no alteration of the source-code is required. Only the messages, culture-dependent elements, are altered by the localisers/translators. These translators also need not possess programming knowledge, as no source code is modified. As a result, there is no introduction of new software errors into the source code of the internationalised program. Furthermore, given the separation between the generic core and the localisable component, the internationalisation team and localisation team can work separately.

Another advantage of the internationalisation and localisation process is that there is a faster product-to-market time for the internationalised version compared to the non-internationalised program. This faster time is because the non-internationalised program source code has to be modified, and recompiled each time a new market is targeted. If the non-internationalised software needs a total re-construction, the time required to rebuild the software will be even longer

as the software may have to be redesigned and implemented. With the quicker time-to-market, users in the international markets will be able to obtain the localised software, which is technologically the same as the original software, at nearly the same time as the original software is available.

The internationalisation and localisation process results in only one internationalised component/generic core, together with a description of the necessary culture-dependent component; this can be used to produce localised components for each of the target cultural groups. As mentioned earlier, the culture-independent component/generic core contains the bulk of the functional source code. Updates to the software would only require update of that generic core. In comparison, retro-fitting results in many program versions. There is a different program version for each different cultural group. Updating is difficult, as developers need to update all these different versions.

Furthermore, the reduced effort in localising to the different target markets translates to reduced-costs for each localised product (Hall and Hudson, 1997). With the reduced costs and relative ease of localisation, more software for other cultural groups may be developed.

In sum, software adapted by internationalisation and localisation is better than software adapted by retro-fitting in terms of ease and efficiency of developing software for many cultural groups. As the focus of this thesis is the development of software for different cultural groups, literature on the development life-cycle of software for the global community is described in the next section.

2.5 Review of Selected Internationalisation and Localisation Literature

This section provides a review of literature pertaining to approaches and activities in developing software for multiple target cultural groups. In other words, this section provides a review of published works that concern the global-software development life-cycle. As mentioned in Section 1.4, global-SDLC is a conventional software development life-cycle (SDLC), which includes internationalisation and localisation. A conventional SDLC consists of the

requirements analysis, design, implementation, testing, and usability evaluation phases.

Trollip and Brown (1987)

One of the earliest published works of re-engineering software for multiple cultural groups was reported in Trollip and Brown (1987). Trollip and Brown (1987) wanted to access the international multi-lingual market. They also wanted to translate an examination software and delivery system called *The Examiner*. The original version of *The Examiner* had a US English interface. Trollip and Brown (1987) wanted a set of computer programs to handle all target languages, and required that translations could be carried out by a person who did not know programming. As part of their solution, all the hard-coded messages of text visible when the software is running within the main program were removed. Trollip and Brown (1987) placed the software's visible text in a separate database. This separation means that a different language version of the program can be presented using the same program but with a different text database. They also developed a simple editing system which allowed the translator to edit the text that appears on the screen. The translation was then checked to ensure that the displayed translated text was from the correct text database during program execution. Trollip and Brown (1987) successfully completed the system in Dutch.

This paper provided succinct details about the internationalisation process as well as localisation process, although the terms "internationalisation" and "localisation" were not used at that time. The only contention with the article was that Trollip and Brown (1987) did not provide any information about whether the users of the software had any problems with the translated software since no usability evaluation of the translated software was provided.

Nielsen (1990a, 1990b, 1996)

Jakob Nielsen's (1990a) edited book was probably the first collection of articles concerning the provision of software for the international market. Nielsen's (1990b) paper is one of the seminal papers on usability evaluation of localised/translated interfaces. In that paper, Nielsen (1990b, p.39) claims "an interface which is used in another country other than the one where it was designed, is a new interface." This interface, used in another country, could be the original interface or a translated interface. Nielsen (1990b) also states that

usability studies conducted in the software's country of origin are not valid elsewhere and that the software must be tested in the target countries.

Another interesting point about language translation and testing was made in Nielsen (1990b). He made this observation in usability evaluation of software with interfaces translated into Danish. Nielsen found that a translated menu item which is perfectly adequate when seen by itself, may not work when the menu item is used as part of dynamic interactions. Thus, the interface must be evaluated to determine the effectiveness of the translation.

Nielsen (1990b) concluded that a carefully developed user interface may be debased by poor language translation and that usability evaluation of the translation should be conducted. In particular, Nielsen recommends using translators with knowledge of human-computer interaction principles for dialogue design, and employing real users from the target population to test the interface.

Nielsen (1990b) also states that translation of the interface may not be sufficient. He provides an example that supports his assertion: LYRE, a French hypertext system for teaching poetry, allows students to see the poem from various viewpoints. LYRE allows the teacher, but not the students, to add new viewpoints. This design is acceptable to Southern European tradition. An alternative design allowing students to make changes is unacceptable, as the teacher's authority would be undermined. However, Nielsen states that people in Denmark, where Scandinavian attitudes are prevalent, would not accept the current design of LYRE as the system limits the students' potential for independent discovery. Thus, translation of the hypertext system would not be enough. In this case, the systems would probably need to provide two modes: one mode which allows the addition of view points, and another that does not allow the addition of view points.

In Nielsen (1996), he provides further guidelines on how to conduct international usability tests. He suggests conducting international inspections, which involves people from the target countries looking over the user interface and determining whether the interface would cause any problems in their country. In particular, Nielsen (1996) advises two fundamental parts of international user testing: to involve real users, and to have them do real tasks with the system without getting any help.

The argument that "each interface when used outside its country of origin" is a "new" interface is valid given that software developers would not know how the software will be used in the target population, as exemplified in the LYRE example. By employing representative users from the target populations, software developers will be able to determine whether users will face problems using the "new" interface. Surprisingly, Nielsen (1990b, 1990a) made no mention about the usability tools employed to evaluate the "new" interface. Paraphrasing his argument, the original usability assessment tool when used by another cultural group (and not just by another country) other than the one it was designed for, is a new usability evaluation tool. As such, the usability assessment tool would also need to be evaluated in the new context.

Del Galdo (1990)

Another article from Nielsen's (1990a) book was Del Galdo (1990). Del Galdo provides guidelines to design software products for the international market. She observes that a product was adapted for international markets after the software had been finalised in the original language version. She argues that if any changes such as capabilities to present text from right to left or vertically were needed in the software, the changes would require much effort. It may even require a rebuilding of the software. Del Galdo says that local conventions must be considered early in the design stage of the product and that product testing must be performed with users from various target cultural groups. Del Galdo also provides categories of items that need to be considered when translating the language of the software's interface. These categories include character sets, various collating sequences, the ability to display international date and time formats, and the ability to present text from right to left or vertical languages. She also advises that the target population and the message to be conveyed must be considered first before using icons and symbols. By taking into account all these local conventions, the conversion would be more effective and more efficient.

The guidelines and advice provided by Del Galdo are sound. However, the suggestions she provided refer predominantly to European, British and North American markets. This focus is hardly surprising given that the software market at that time was mainly in those areas.

Sukaviriya and Moran (1990)

One of the notable results from the experiment reported in Sukaviriya and Moran (1990) related to the user's language background. The study investigated the person's language preference. Both English and Thai have a common grammatical structure; both languages employ an action-object pattern, for example, "Move the chair" and "Delete the file", whereas the Indian subcontinent languages employ the opposite object-action pattern. The results of the study supported the hypothesis that a subject's choice of an object first, and then an action on the object, or vice-versa, was dependent on his or her language background. This evidence confirms that a person's cultural (linguistic) background may impact on how they interact with computers.

Taylor (1990, 1992)

As little had yet been written about how to conduct internationalisation, Taylor (1990) describes the internationalisation process using the Hewlett-Packard Native Language Support System. He illustrates the internationalisation process using a simple C program. The data output of the program was in Spanish as well as in French. As part of the internationalisation process, he provides examples on how to extract the text messages and language-dependent data. The language-dependent data were placed into a message catalogue. Taylor also provides suggestions on how to convert the program to handle Asian languages.

In Taylor's (1992) book, a more detailed description of the internationalisation and localisation process is given. He describes three approaches: compile-time, link-time and run-time internationalisation. In compile-time internationalisation, the culture-dependent messages in the source code are altered when a new language is required. The modified source code is then compiled, linked and run. There is a different executable for each language version of the software. In link-time internationalisation, culture-dependent components are isolated and compiled separately from the generic core. To obtain the executables which allow interactions in a particular language, the appropriate object file of a language is linked with object files of the generic core and run. In run-time internationalisation, the culture-dependent components such as menu commands are kept in a message file external to the program. Each message file will contain the commands in a particular language. During run-time, the software can allow interactions in different languages depending on which

message file was selected. In compile-time and link-time internationalisation, there will be as many executables as there are language versions of the software. However, run-time internationalisation results in only one executable. Taylor (1992) was probably the first detailed document that provided an account of the generic implementation of internationalisation and localisation. The latter publications were targeted at internationalisation and localisation of software for their corporation's operating systems. Taylor's (1990, 1992) papers focused mainly on the design and implementation aspect of the internationalisation and localisation of the software. Other phases of the global-SDLC were not provided.

Kennelly (1991)

Digital Equipment Corporation (Digital) provides details on how to develop international software on Digital's operating systems in Kennelly (1991). In the book, Kennelly describes the International Product Model. This model is employed to provide a framework for the various groups of people involved in development of software for international markets. The International Product Model comprises:

- The International base component: This component is sold worldwide without modification and contains the executables, images, internal files, and command procedures.
- User interface component: This component contains the language and text processing component as well as message files, command menus, and command procedures with text.
- Market-specific component: This component is added to meet special requirements of a specific region that shares a language and set of cultural conventions. It includes keyboard maps, telecommunication controls, and printer controls.
- Country-specific information component: This component contains the set of required documentation for selling the product in a specific country; for example, licence certificates, warranty information, and product description.

This model is similar to the model described in Madell, Parsons and Abegg (1994) in Section 2.4.2.

The contents of this book include issues pertaining to the data conventions, language, dialect, keyboard layout, data input and display conventions, as well as collating sequences. These issues have been identified by Del Galdo (1990).

Kennelly (1991) addresses only cultural elements at the symbols level that relates to character sets, language, and data format conventions. Deeper issues such as rituals and values are neglected.

Uren, Howard and Perinotti (1993)

Uren, Howard and Perinotti (1993) provide mainly information about internationalisation and localisation for Western European languages on IBM PCs. While Uren, Howard and Perinotti (1993) do not provide technical details of the internationalisation and localisation process, they furnish detailed guidelines about translation, documentation, as well as validation and testing. The guidelines also focus on the accuracy of the translation and concerns about whether the software is performing correctly. For example, to improve accuracy of translation, Uren, Howard and Perinotti also suggest conducting back translation; that is, translating the script or text back to the original language by employing a different translator, and then comparing the original with the back-translated work. They suggest that testing should be conducted with non-American equipment in the non-American environment, that is, in the environment of the target cultural group. In the book, Uren, Howard and Perinotti assume the software is developed in the US and provided for the rest of the world. Uren, Howard and Perinotti focus mainly on the design and implementation phases of the global-SDLC.

Russo and Boor (1993)

Russo and Boor (1993) gives more information about the cultural differences with regards to image acceptability and image recognition. They re-iterate the importance of addressing a culture's impact on the design at the beginning of the product development cycle. They also suggest the need for development teams to establish a close working relationship with the target cultural groups. Russo and Boor (1993) advocate conducting reviews of information and feedback provided by the target cultural groups at various stages of the product cycle. Thus, any omissions of cultural factors that need to be considered in the design would be identified earlier rather than later in the cycle.

Russo and Boor (1993) also suggest that usability tests should be conducted where possible at the same time as the domestic usability tests, before the product is released. Representatives of the target cultural groups can assist in identifying the subtle elements of an interface that may be confusing or offensive. By testing early in the product cycle, the results from all target cultural groups can be consolidated and applied into the development of the product.

Russo and Boor (1993) consider the conceptual level of internationalisation and localisation. They also highlight the importance of ensuring feedback throughout the cycle. However, the recommendations provided in the paper were probably based on the authors' observations as no empirical evidence was provided to support their claim.

Nakakoji (1993, 1994), Ito and Nakakoji (1996)

Nakakoji (1993) provides an interesting comparison between the Japanese and American work habits and how these characteristics may impact on the design of software. For example, she describes that employees (in particular, junior workers) are not supposed to object to opinions of their more senior workers especially if more "big" bosses are attending some meeting. Nakakoji (1993) suggests that anonymity in groupware may improve a Japanese group's performance. Nakakoji (1993) also describes Trompenaar's model, which characterises cultural differences. The model consists of six different aspects, which represents a polarity of behaviours or assumptions. She argues that each of the cultural aspects may affect the design of the user interface. For example, one of these aspects is time perception, some people perceive time synchronously, others sequentially. This time perception may impact on human-computer interaction design, since certain groups of people may prefer processing jobs in parallel, while others prefer performing tasks sequentially.

Nakakoji (1994) suggests that developers should look beyond adapting surface-level interface when providing software to a new cultural group. She states that the introduction of software to a new cultural group may reveal many hidden and unpredictable factors. Nakakoji (1994) supports her claim with examples of the American/British word processor. She says that the Japanese have had little exposure to typewriters, common in North American/European countries, and thus notions of cursors, tabs and margins were foreign to Japanese

users. A word count of the document would not work, as Japanese count the number of characters in a document.

Nakakoji (1994) points out that not only language barriers exist when people from different cultural groups communicate. Barriers in the form of social norms and background also exist. For example, she states that if Japanese test users said they liked a system, these users may be telling the truth or may just be too polite to make negative comments about the system. She also suggests that the introduction of computer systems creates additional new cultures. Different disciplines, and different expertise levels, and different roles form their own cultures. Nakakoji (1994) recommends that to successfully introduce a new software system to a different cultural group, software developers must first familiarise themselves with the target culture and then design a completely new system for that cultural group.

Ito and Nakakoji (1996) show how culture can impact on user interface design. They envisage human computer interactions as listening and speaking mode. Listening mode pertains to perception of computer's display by the people. Speaking mode pertains to people giving instructions to a computer system. Ito and Nakakoji (1996) suggest that culture impacts on both the listening and speaking mode. For example, one of the activities in the listening mode is the semantic association whereby people associate semantic meanings to what they perceive from the computer. In semantic association, people associate what they perceive with something that they know. Thus, different semantic meanings may be associated with the same items.

Nakakoji (1993) provides a useful comparison of Japanese and North American work habits. Her examples provide some indication of how software developed for use in one workplace may be rendered less useful in another country's workplace. Her description of the multicultural human-computer interaction design architecture provides a useful framework on how to develop interfaces for multiple cultural groups. While Nakakoji (1993) suggests using the Trompenaar's model as a framework to study cultural groups and describes examples on how the aspects of the cultural model may impact on the design of interfaces, the examples are mainly anecdotal. The advice provided by Nakakoji (1994) to look beyond surface-level issues is valid. After all, culture aspects are

not limited to just language. The model described in Ito and Nakakoji (1996) allows a deeper understanding of how cultural aspects may impact on human computer interaction and user interface design.

Teasley et al. (1994)

Teasley et al. (1994) show that "professional intuitions are not sufficient to design interfaces for culturally-diverse users". Marcus (1993), an expert user interface designer, had proposed three dialogue boxes from a hypothetical word processors targeted at "white American women", "English-speaking European adult males" and "International English-speaking consumers". Teasley et al. (1994) conducted an experiment, employing the representative target users of the proposed designs, to evaluate the designs. The findings from Teasley et al. (1994) show that only 33.6% of the target users selected the dialogue box that was designed for them. The authors conclude that "professional intuition" is neither a sufficient nor reliable method to produce "appealing perceptual experience" in interactive computing systems (Teasley et al., 1994).

This finding shows the importance of obtaining feedback from target users. Without input from representative users, the success or failure of interfaces or software would not be known.

Kano (1995)

Kano (1995) provides technical details about developing international software on the Microsoft Windows and NT platforms. The book details the internationalisation process similar to the process described in Taylor (1992).

Kano (1995) also describes the different levels of localisation. These different levels depend on how important the market is and whether the returns are commensurate with the investment made. In Figure 2.7, Kano (1995) provides some guidelines as to where an English-language user interface is acceptable:

- Small markets (Central and Eastern Europe and Indonesia)
- Markets where your product has no competitors
- Markets where many people speak English (for example, India, Israel, and the Netherlands)
- Markets where the target audience speaks English (for example, scientific, medical, and technology communities)

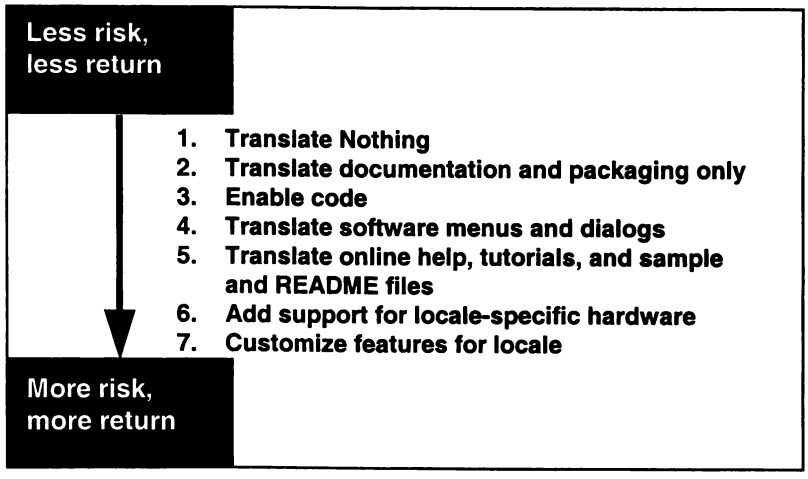

Figure 2.7: Levels of localisation (sourced from Kano (1995))

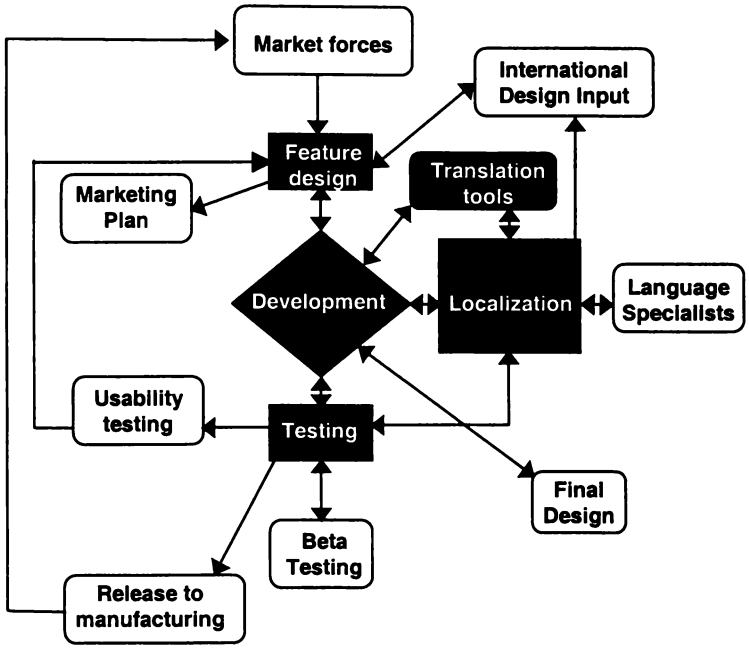

Figure 2.8: Localisation process at Microsoft (sourced from Kano (1995))

Kano (1995) was one of the earliest publications to provide a framework for the global-software development process as well as to describe an international product development process (see Figure 2.8). The figure provides a rough timeline in the development of a product. The design of the product is affected by market forces, usability tests, and development constraints (Kano, 1995). The development team provides resource files, which contain the culture-dependent elements, to the localisation team. The localisation team translates the visible interface commands and messages, resizes the dialogue boxes, and returns the translation back to the development team. The development team recompiles the program with the resources files. The executables are then passed on to the testing team. The testing team reports functionality problems to the development

team, and reports user interface problems to the localisation team. This cycle is repeated until problems are resolved.

Kano (1995) provides comprehensive details on the design and implementation stages of the software development. However, the issues focussed on, cover only the surface level issues of language and character sets. Furthermore, the usability evaluation phase is not elaborated on, despite being shown in her Figure 2.8.

Fernandes (1995)

Fernandes's (1995) book concentrated mainly on design issues of interfaces for the global market. The information provided is based on his experience at Claris Corporation. He provides checklists and examples of the use of languages, visuals, and formats in user interfaces of global software. Fernandes (1995) employed Hofstede's (1994) cultural model to exemplify how cultural characteristics may be employed in interface design.

He provides interesting anecdotal evidence of methods employed in usability evaluation. For example, Fernandes (1995) observed that co-discovery techniques were found to work well in Germany, but was problematic only when one user's command of English was better than the other user's English. He also observed that some users in non-English speaking countries were willing to think aloud in English. Fernandes (1995) also describes his experiences in conducting usability evaluation in Japan. He claims that questions asking how comfortable or how much Japanese like a product were removed because these questions involved feelings and emotions, which the Japanese are not accustomed to talk about. Fernandes (1995) also notes that co-discovery techniques became problematic when people of differing status were put in the room together. Also, women talked very little when they were paired with a man in studies using the co-discovery technique.

While the book provides many examples and guidelines, all information is based on his personal experience. His observations need to be confirmed by empirical studies.

Dray (1996), Dray and Mrazek (1996)

Dray (1996) provides suggestions based on her experience and observations as a consultant. She restates the need to evaluate the usability of a

localised design to ensure that the design and localisation are effective and not offensive to the target cultural groups.

One of her significant suggestions is to localise the test plan and the test protocol. For example, the tasks provided should be altered so that they make sense to an international tester. The usability evaluator must be aware of local customs when setting up the evaluations. The testing situation may also require modification. She believes that evaluations will be more successful if they are carried out by a local partner, rather than by a foreigner. She suggests that in some countries a woman evaluator may be most effective, whereas in others it would be more appropriate for her to observe from another room.

Her suggestions to localise the test plans, and the methods employed in the international testing appear to be valid. However, the suggestions provided are probably based on her own observations and have not been validated by further studies.

Dray and Mrazek (1996) describe how they collected data for Hewlett Packard as part of its new global product development. The data was collected using *contextual research*, that is, by observing families using their computer in their own homes in six locations in three countries, namely, Germany, France and the US. Due to the cultural differences, the US approach had to be slightly altered. As Dray and Mrazek (1996) were from US, they recruited women translators in Germany and France as part of the European team. Based on their experiences, Dray and Mrazek (1996) advise usability evaluators to rely on the knowledge of local partners on how to behave during visits, as well as in guiding the recruitment of participants. Dray and Mrazek (1996) also advise focussing on forming a rapport with the target group on the first visit and maintaining this rapport throughout the study.

Dray and Mrazek's (1996) chapter provides information about data collection; it shows that they had adapted standard approaches to suit the target population as well as obtaining assistance from local partners for the study.

Hoft (1996)

Hoft (1996) provides a way to study different cultures. She suggests employing a cultural model as a framework to compare the cultural differences. The cultural model uses international variables, which are categories used to

organise information about cultures. The models can be employed to collect data that allows designers to create designs that closely match the needs of the target cultural groups. The cultural data collected can be employed to develop cultural profiles of groups of users. This cultural profile of target users can then be applied to design, test and evaluate products for the target groups. For example, if the profiles are quite similar, then the product need not be customised.

This paper provides a useful way to collect information about target populations. The cultural profiles information can then be employed in the internationalisation of the software. The information would also indicate the cultural issues that need to be addressed when designing the generic core.

Herman (1996)

Herman (1996) reports on a usability evaluation of a real system in Singapore, whereby objective evaluation results from performance measures correlated poorly with subjective evaluation results from a questionnaire and an interview. This poor correlation was illustrated with a case in which a participant actually broke down and cried during the software evaluation session but the participant was very positive about the software in the post-test interview. Herman (1996) attributed this behaviour to the Eastern culture whereby it is "considered culturally unacceptable to criticise the designer directly or openly, as this may cause the designers to lose face" (Herman, 1996). Similar observations were also made in a usability evaluation of public information kiosks in Singapore (Lim and Usma, 1998).

One contention with Herman's (1996) observation is that the observation is anecdotal. The cause of the behaviour was not ascertained. It does, however raise questions about the validity of employing usability assessment tools from developed nations; tools which may not be appropriate when employed in the countries outside the tools' origin.

Hall and Hudson (1997)

The following sections review the relevant works from the book edited by Hall and Hudson (1997). Hall and Hudson's (1997) book probably contains the most comprehensive literature on implementation and practical aspects of internationalisation (i18n) and localisation (l10n) of software.

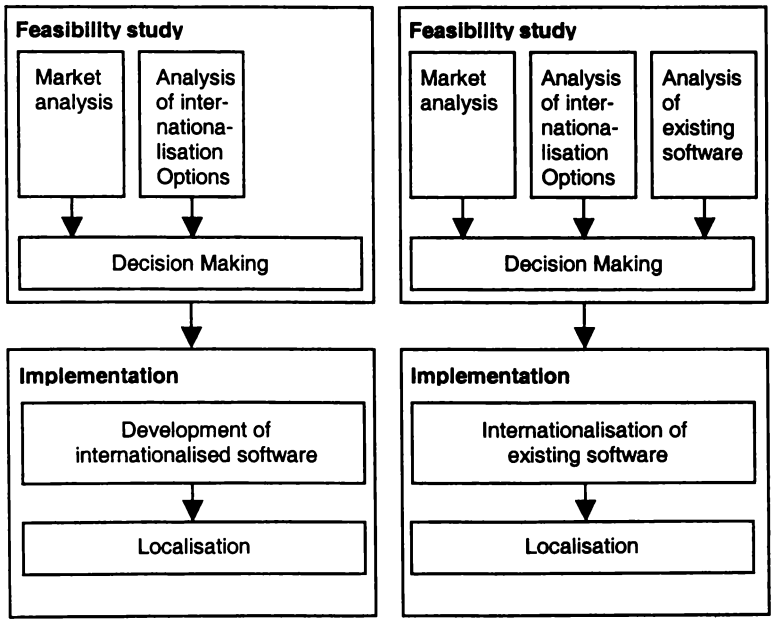

Figure 2.9: Ideal Model of i18n & l10n (sourced from Honkela et al. (1997))

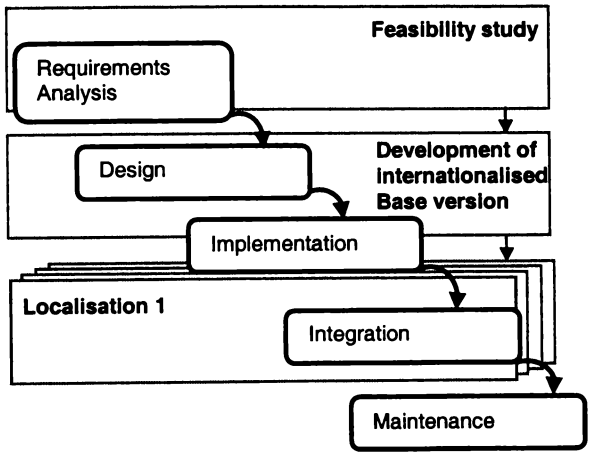

Figure 2.10: Waterfall model and i18n/l10n (sourced from Honkela et al. (1997))

In Chapter 3 of Hall and Hudson (1997), Honkela et al. (1997) recommend a model of a life-cycle for developing software for the international market. This model is based on approaches summarised in Hudson, McHugh, and Kalpakas (1997). The model can be employed to produce localised products either from the beginning or re-engineered from an existing product (see Figure 2.9). The software development model is divided into two phases, a feasibility study, and implementation phase. These phases are compared with the classical waterfall model in Figure 2.10.

Feasibility Study

Software developers usually conduct a feasibility study first to determine whether it is worthwhile to internationalise and localise the product to the target markets identified (Honkela et al., 1997). Among issues that are considered in the

feasibility study are: the market needs of the target group, the costs and benefits of internationalisation and localisation, internationalisation options and whether to internationalise and localise or just localise the product. These issues are examined before any decision is made to continue with the internationalisation and localisation of the software.

Development of Internationalised Base

Three implementation activities are involved in this phase. These activities include internationalisation, with implementation activities, localisation and quality assurance (Honkela et al., 1997).

Internationalisation includes designing and modifying the software so that culture-dependent elements are placed into message catalogues. The design of the software takes into account the requirements and specifications of the multiple cultural groups targeted. In the case of software that already exists, internationalisation will still take place, although some major re-engineering may be required if culture-dependent components are "deeply embedded" in the program. The chapters by Lehtola et al. (1997) and Kokkotos and Spyropoulos (1997) present examples on how to design internationalised software.

According to Lehtola et al. (1997), localisation involves the provision of documentation, message files, device drivers and other elements for the target cultural groups. Localisation can be carried out by developers of the internationalised software or they can be conducted by people from target cultural groups. With regards to localisation, the biggest task required during this phase is translation (Lehtola et al., 1997).

McHugh, Honkela, and Hudson (1997) in Chapter 12 of Hall and Hudson (1997) furnish information about testing and quality assurance. They also provide guidelines on the evaluation of the success of translation, the measurement of quality assurance for localisation, and the assessment of the effects of software internationalisation on overall system performance. For example, when multi-octet character codes are used, large amounts of memory is required and this demand may reduce system performance.

The chapters in Hall and Hudson (1997) provide the most detailed information of a model on the development and adaptation of software for the

global market. The stages are described in detail and the information provides a useful guide at the conceptual as well as at the practical level. Surprisingly, the usability evaluation phase is missing and there is no mention of employing target users to evaluate localised products. Hall and Hudson (1997) only describe testing of the functionality and performance of the localised product.

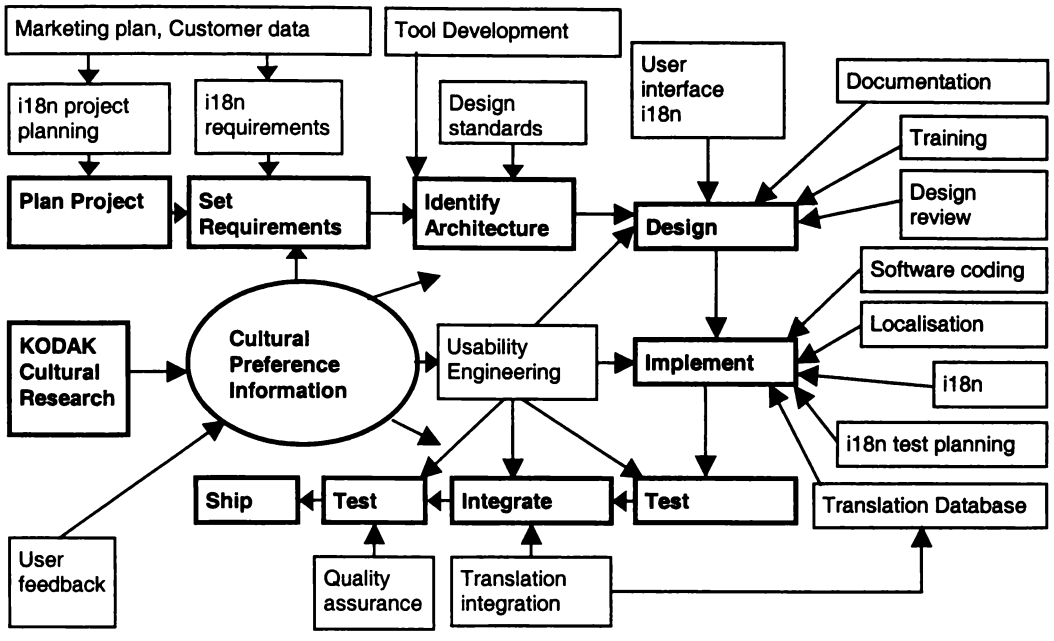

Figure 2.11: i18n and l10n at Kodak (sourced from Prabhu et al. (1997))

Prabhu et al. (1997)

In this paper, Prabhu et al. (1997) present the generic product development process at Eastman Kodak Company (see Figure 2.11). These activities included in the process are: plan project, set requirements, identify architecture, design, implement and test, integrate, test and ship. Central to this generic development process is the cultural preference database. The database contains information required for the internationalisation and localisation of products at Eastman Kodak Company. Examples of information required include user interface design issues, preferences pertaining to colour, and interaction styles for different cultural groups. Prabhu et al. (1997) describe a methodology employed at Eastman Kodak Company to study and understand users' needs and preferences of internationalised versus localised products.

By having that cultural information database, the company would have a central repository of information available to Kodak employees. Little was mentioned about the development cycle as a whole. The figure in their paper (see

Figure 2.11) does include usability engineering, which one assumes to include testing of software with the target population.

Evers and Day (1997)

Evers and Day (1997) conducted a study to examine users' culturally specific design preferences. Data was collected from 206 international students from three main groups: Indonesians, Chinese, and other Asians – students from countries such as Taiwan, Singapore and Japan. Australian students were recruited as the control group. Results indicate that design preferences do affect interface acceptance. Chinese found usefulness a distinguishable variable, whereas Indonesians find ease of use more important. According to the authors, "The results suggest that Chinese will try to work with a useful interface even if it is hard to use, whereas Indonesians will tend to give up easily when an interface is hard to understand.". This study provides evidence that the different cultural groups have different preferences which could impact on the user interface design.

Day (1998a, 1998b, 1998c, 1999)

Day (1998a) and Day (1998c) contained special issue papers for "Shared Values and Shared Interfaces: The Role of Culture in the Globalisation of Human-Computer Systems". The papers in the two journal issues range from research on the design of interfaces to the use of different methods to develop multi-lingual interfaces. One of the articles, Carey (1998), is reviewed later in this chapter.

In his editorial, Day (1998b) notes a number of issues that are still relevant today. He points out that most HCI-research is dominated by the developed nations of North America and Europe. Day (1998b) also notes that the diverse contexts in overseas locations may call into question the results of studies conducted in developed nations. He also notes that much research – in particular state of the art on usability – is proprietary. Day (1998b) emphasised the need for "Western" dominated software companies to address the concerns raised.

Day (1999) provided a fascinating story to support the main points of a treatise in his paper:

> *"Technology is a tool, nothing more. As an extension of human abilities, it must be appropriate within the context of use. Otherwise, it may be more an impediment than an aid in advancing human goals and objectives."*

and

"Culture is the ultimate context of use."

The points he makes are exactly right. Day (1999) states that technology transfer in terms of internationalisation is the porting of product from a cultural group in which it was developed, to another cultural group. Day (1999) contends the technology transfer is not a success until the technology is used to its full intended extent in the new cultural context. He states that there are many examples where technologies had failed because the intended users "refused" to adopt them. Day (1999) points out that cultural incompatibility is the main reason for the failures. Day's comments are valid. If the technology cannot be used in the target context, then the technology transfer is considered unsuccessful. Countries that export technology must ensure that the technology they export is appropriate or adapted appropriately for the target market. Otherwise, both the exporters and importers will suffer; the exporters in terms of loss of sales, and the importers, an inefficient and ineffective tool. The term "technology" referred by Day (1999) is deemed to apply to software production. Software exported to multiple cultural groups must be adapted appropriately for the target groups.

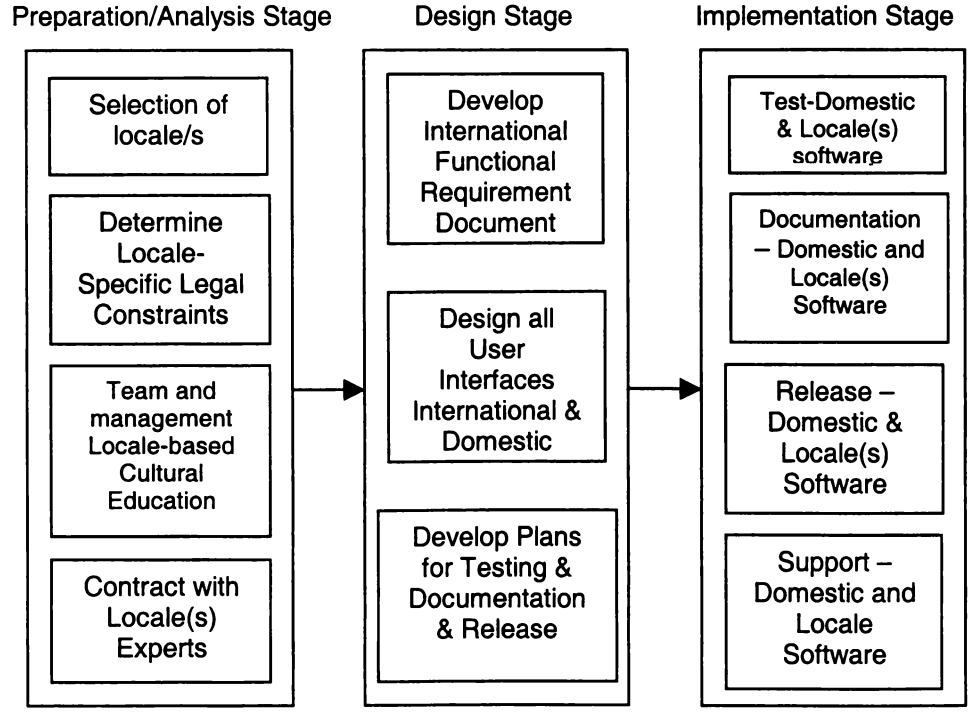

Figure 2.12: Stages in global software development (sourced from Carey (1998))

Carey (1998)

Carey (1998) derived a framework of the stages in the international software development from the 10 books she reviewed. She contends that developers following the stages provided in Figure 2.12 and adhering to the specific activities provided from the books, would enhance the likelihood of success for global software products.

This global software development framework appears to be an adequate model for developing global software. However, the framework does not include usability evaluation, probably because the books which were reviewed did not contain substantial work on usability evaluation, if any at all.

Honold (1999, 2000)

Honold (1999) employed focus groups, usability tests and questionnaires to gather information about how Chinese and Germans learn to use cellular phones. The focus groups method was modified for use in China. For example, in China, Honold (1999) and a Chinese moderator accounted for potential problems resulting from "face saving" and status differences. To counter these potential problems, users of a similar age and similar profession were chosen. Scenarios for the usability testing were also modified. For example, the German users were asked to make a call to send birthday greetings, whereas the Chinese users were asked to make a call to congratulate the family as the son had passed school-leaving examinations. The studies showed differences in how Chinese and Germans learnt how to use the phones. Chinese relied more on informal and oral information, whereas Germans' main source of information was printed user manuals.

Honold (2000) employed empirical studies to collect qualitative data to identify factors that influence the use of products in different countries. In particular, interviews and observations were used to study the use of a German-developed washing machine in Indian households. The Indian employee, who accompanied the author in the study, adapted the interview situation to Indian conventions.

While the results of the studies were important, the fact that the author had adapted the methods to the situation by consulting a local moderator was

equally important. It would have been interesting to see whether un-adapted methods would have also worked.

Bourges-Waldegg, Moreno, Rojano (1999)

Bourges-Waldegg, Moreno and Rojano (1999) describe a study carried out in Mexico. The study was conducted to observe and evaluate the usability of an educational software application. The application was used for teaching mathematics and science. The interface of the software was in English and Spanish. One of the important issues raised in the paper was that usability should be evaluated according to educational goals and these goals may vary from cultural groups to cultural groups.

Kurosu, Motoki and Itoh (2001)

Kurosu, Motoki and Itoh (2001) used both US and Japanese usability guidelines to rate US and Japanese websites. Fifteen US sites were evaluated according to 118 guideline items obtained from published US literature. Each of the sites was scored against the 118 guidelines.

Fifteen Japanese sites were evaluated according to 65 Japanese guidelines. These Japanese guidelines were selected, from the 118 US guidelines, based on importance by 15 Japanese participants.

It was found that the Japanese web sites rated better than the US web sites on both Japanese and US guidelines. The Japanese sites appear to have more impact than US sites. However, the authors who conducted the evaluation of the sites, admitted that there might have been some bias in the evaluation. They suggest that a counter experiment using US subjects should be done.

It was interesting to note that US guidelines were adapted by the authors to suit the Japanese context. Assuming that Japanese sites do have more impact than US sites, it would seem that the US and Japanese guidelines were interchangeable. The guidelines imported into the new context appear to perform similarly to tools that were adapted to the new context. Besides conducting the counter experiment using US participants, it would be useful to identify what guidelines Japanese web site designers find important in designing web sites, and use the guidelines to rate US websites.

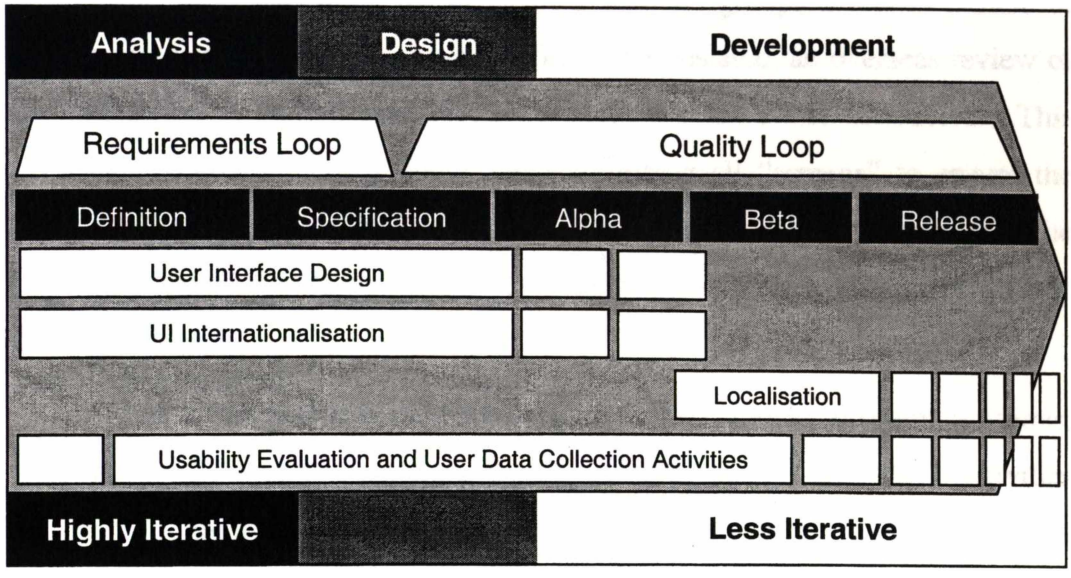

Figure 2.13: i18n & l10n at Hyperion (Livermore & Coronado, 2001)

Livermore and Coronado (2001)

Livermore and Coronado (2001) described their experience in integrating internationalisation and localisation activities into the software development process at Hyperion. Hyperion is a US-based software company which provides business analysis software for large corporations world-wide. Some problems experienced by Hyperion were that their product was not translated in a timely fashion, and internationalisation was not considered at the beginning of the product development process.

The effect of these problems was minimised by the formation of the User Centred Design (UCD) group. This group was successful in incorporating new practices into the development process including early assessment of customer needs in terms of functionality and usability. The UCD experts conducted internationalisation activities, which took into account the impact of translation of text on size and layout of forms, and dialogue boxes. They also investigated the meanings of icons and symbols used in the different versions of software targeted at different markets. Guidelines for design layouts and controls which considered internationalisation requirements also resulted from UCD group's activities. It is interesting to note that the UCD group also identified and used international "variables", such as Hoftede's (1994), in the design process.

Livermore and Coronado reported that the localisation process started at the end of the product's first alpha cycle (see Figure 2.13) where the user interface is at least 90% complete. The development team gives "translatable" files to the

localisation team. Also, an initial meeting with all groups involved is held to discuss issues of translation. After the files are translated, an overseas review of the translation is carried out. Finally, a linguistic review is conducted. This involves a translator sitting down and scrutinising all "screens" to ensure the translation is appropriate for the target context. Other documents handed to the localisation team for translation include help files, HTML files, and printed document files.

The UCD practices, and internationalisation and localisation activities, were integrated into the software development process. The UCD activities included obtaining executive support, educating those involved in the development process as well as updating various groups on the latest technologies in internationalisation and localisation. Little information, however, was reported about what was specifically done in the internationalisation and localisation process. Again the internationalisation and localisation process only addressed linguistic and graphical issues.

Onibere et al. (2001)

This paper reports on research carried out to determine whether a localised interface is preferred by users in a multicultural and multi-lingual country where a non-local language is nationally used. Onibere et al. (2001) aim to determine the effect of culture on the understanding of commonly used phrases/jargon and icons in Botswana. They hope to find answers to the following questions:

- Are local symbols more acceptable to users as icons?
- Do users understand clearly the phrases/jargon currently/commonly used?
- Would users prefer a localised interface?

Onibere et al. (2001) used surveys and interviews to collect information.

Almost all interviewers were Batswana. [A Botswana citizen is a Motswana; plural, Batswana.] Fifty-seven per cent of the 324 users surveyed would like localised software for Botswana, but only 25% wanted Setswana, the national language, for commands in the localised software. The authors reasoned that the participants' reluctance was because the national language is not the

native tongue of all the participants. Their findings also suggest that there is no need for localised icons. However, this finding may be due to the fact that all the participants' had used software with an English interface, and thus were familiar with non-localised interfaces and icons. It was surprising that only people who had computing experience were recruited, given that their exposure to US software may have biased their opinions.

2.6 Summary

This summary is organised according to the global-SDLC, that is, requirements and design, implementation, testing and usability evaluation. Internationalisation and localisation stages are reported in the design and implementation sections respectively.

2.6.1 Requirement Specifications and Design

Before the software can be internationalised, software developers must know what the culture-dependent elements are, so that these elements can be isolated from the software.

Many references, such as Hall and Hudson (1997), Fernandes (1995), Kano (1995), O'Donnell (1994), Uren, Howard and Perinotti (1993), Russo and Boor (1993), Taylor (1992), and Del Galdo (1990) list the numerous factors that may impact the design of the internationalised software. These culture-dependent factors are classified in the following categories.

- Language: character sets, reading direction (flow), sorting sequence, punctuation marks;
- Data display formats: date, time, currency, numbers, telephone numbers, address formats;
- Calendars, weekends, day turnovers;
- Units of measures: paper size;
- Images/visuals, colours, sounds; and
- Regulations, standards which must be complied with.

This list is not exhaustive. In addition to these culture-dependent factors, many guidelines and suggestions are also available in the references noted above.

One contention raised by Bourges-Waldegg and Scrivener (1998), in Day (1998a), about the guidelines provided in the available literature is the generality of the guidelines employed to develop intuitive user interfaces for the global market.

These guidelines are *general* guidelines and are not *specific* to any particular context. For instance, "Colours have different connotations in different cultures" (Del Galdo, 1990) or "Use appropriate and familiar objects" (Fernandes, 1995), "Be sure that gestures and images of the body in the human interface are appropriate for the target culture" (Apple, 1992). While these guidelines are to a certain degree helpful, specific information would be of more use. For example, "In Malaysia, colours associated with festivities include red for the Chinese, and green for the Malays.". Only when software developers have more specific information, can they develop software that is acceptable to target cultural groups. These specific guidelines are especially important when the developers lack the time or resources to collect information from the target cultural groups.

One way to collect information about target cultural groups is through the use of cultural profiles of target groups of the software. Nakakoji (1993) discussed the use of Trompenaar's "cultural aspects" to provide information about the users. This information can be applied by designers in the development of software. Cultural models were also proposed by Hoft (1996). She provided details on how to create cultural models, and collect information required for the design and internationalisation of the user interface. An example of an application use of a profiling method is documented in Prabhu et al. (1997).

Besides creating a profile of the target cultural groups, some researchers are employing qualitative methods to collect information about how target groups use products. For example, Honold (1999) employed focus groups to study how Chinese and Germans learnt how to use mobile phones, and Dray and Mrazek (1996) employed contextual research to study how families used computers in their homes. By identifying how target cultural groups use products, strategies to improve the application of the products can be identified and implemented. It was interesting to note that Dray and Mrazek (1996) and Honold (1999, 2000) adapted their methods to suit the target market they were studying. The adaptation was recommended by their local partners in the study to improve the effectiveness of

the methods employed. The recommendation provided by Nielsen (1996) to hire local consultants appears to be sound.

With respect to the design of the interface, articles such as Del Galdo (1990), Russo and Boor (1993), Fernandes (1995) and Dray (1996) give information on the design of the interface for the international market. Some authors suggest creating culture-neutral interfaces, whereas others suggest creating localised software from internationalised designs which are devoid of cultural elements (Fernandes, 1995). It is recommended that localised interfaces be tested by the target users.

2.6.2 Implementation

After the design stage, the generic core of the software design is implemented based on the systems design and specifications. Uren (1998) presents a list of books and articles, sorted by the myriad operating systems such as Microsoft Windows, Macintosh, and flavours of UNIX, which cover the programming aspects of internationalisation and localisation. Today many major computer organisations, such as Microsoft (Schmitt, 2000; Kano, 1995); Apple (1992), Digital (Kennelly, 1991), Hewlett Packard (Madell, Parsons and Abegg, 1994), Sun Microsystems (Tuthill, 1993), and X/Open Company (1993), provide similar implementation recommendations in the development of global software. For instance, these publications supply information that accommodates the target cultural groups' language and its language-associated issues, such as character sets, sorting, character display, and the data display formats. Details of how the major software companies approach internationalisation and localisation are also available in Hall and Hudson (1997). By consulting the aforementioned resources and those presented in Uren (1998), software engineers should be able to develop software that runs on the various operating systems.

Current software can accommodate different languages and its language-associated elements. For example, the software allows users to change elements such as data display formats as well as input methods. Presently, software such as Macintosh OS X, Microsoft Windows 2000 and Office 2000 allow the different languages to be used in one document under one operating system, as opposed to having multiple operating systems for each language.

Software which can accommodate different languages appeared in the Xerox "Star" in the early 80s and allowed users to interact with the computer in different languages (Johnson et al., 1989). Surprisingly, about two decades later, today's software corporations are still tackling problems related to providing software that allow interactions in different languages. The internationalisation and localisation literature is still predominantly concerned with pre-composed messages. This observation is in evidence not only from published works, but from numerous internationalised software packages which cater only for language and cultural elements such as currency, date, time formats as shown in Section 2.6.1. These cultural elements fall into the "symbols" category, which is located at the surface of the cultural model in Figure 2.1.

Nakakoji (1994) pointed out that deeper cultural issues in software development, such as values, were ignored. Today, the deeper cultural issues are still ignored. Software applications that cater for these deeper issues are not likely to materialise unless the large profit-driven software companies believe it is economically feasible. While software applications are available in many languages, software companies have yet to release software that incorporates deeper cultural factors.

2.6.3 Testing

With regards to the internationalisation and localisation processes, activities in this phase relates to the accuracy of the localisation that is the translation and performance of the software (McHugh, Honkela and Hudson, 1997). Comprehensive lists of the testing procedures can be obtained from Kaner, Falk and Nguyen (1993), Luong, Lok and Driscoll (1995), and McHugh, Honkela and Hudson (1997).

2.6.4 Usability Evaluation

Nielsen (1990b), Russo and Boor (1993), and Dray (1996) supply information on how to conduct usability evaluation of adapted software in the target cultural groups. However, the authors neglect to examine the effectiveness of the extant usability assessment tools and techniques when used in the target cultural groups. It is assumed that these usability assessment tools and techniques, which predominantly originate from the developed nations, are used and will work in target markets such as Asia, a potentially huge market for localised software

and of increasing economic importance to the software companies of developed nations. This assumption is incorrect as supported by the examples of anecdotal evidence from Herman (1996), and Nakakoji (1994) and Fernandes (1995). These tools, which are widely used in the developed nations, may not be appropriate when used outside these nations, such as target cultural groups in developing nations.

Furthermore, the information available on UATs is based on authors' experiences rather than an evaluation of the UATs. For example, Modica and Fiedler (1999) states that the tips and suggestions on evaluating localised user interfaces were based on their experience in conducting usability evaluation in non-English speaking countries. Further examination of the literature reveals that, reported works on the usability evaluation of localised software conducted in countries outside the US, are also mainly based on anecdotal evidence. For example, Fernandes (1995) illustrates usability testing conducted in Japan and Germany. However, the examples provided were observations made by him. One such observation was that, in Japan, the co-discovery technique was found to be problematic when people of differing status were employed; in particular, women when paired with a man were found to talk very little (Fernandes, 1995). Also, work reported in Herman (1996) and Lim and Usma (1998) was anecdotal, as the main concern of their studies was not on the UATs per se. Lim and Usma (1998) were evaluating usability of public information kiosk whereas Herman (1996), was evaluating a software application created for a group of professionals (with little or no human factors input).

2.6.5 Small Cultural Groups

Another point of contention is that much of the internationalisation and localisation literature originates from developed nations such as the US, Japan (Fernandes, 1995), and developed nations of Europe for instance, France and Germany (Dray and Mrazek, 1996) and the United Kingdom/Europe (Hall and Hudson, 1997). As the studies are conducted in these developed countries, the results from these studies only be applicable to these nations. More research is being conducted in developing countries, especially India and China (Honold, 2000; 1999) perhaps because these two countries have huge market potential. India and China between them comprise a third of the world's population.

Minority ethnic groups and small developing countries are neglected, given the lack of profitability and poor return on investments from these countries. It is normally left to the locals of those countries to conduct research, for example, Onibere et al. (2001) in Botswana.

2.6.6 Incomplete SDLC

The majority of the available reports do not cover all phases of the software development life-cycle as well as internationalisation and localisation. For example, many of the literature sources, such as Fernandes (1995) and Russo and Boor (1993), cover only the requirements/user interface design phase; Hall and Hudson (1997), Kano (1995) and Uren, Howard and Perinotti (1993) report on design, implementation and testing, whereas Nielsen (1990b) and Dray (1996) deals only with the usability evaluation phase. However, with the exception of Livermore and Coronado (2001), no literature was found that documents the software development process – which incorporates internationalisation and localisation – from the inception of the design right through to usability evaluation.

Another omission in the literature is the lack of information on the inclusion of internationalisation and localisation activities into a software development life-cycle, which can then be employed to provide software for multicultural groups. Other than works such as Hall and Hudson (1997), current literature provides little or no information about internationalisation and localisation in relation to the software development life-cycle. While Kano (1995) (see Figure 2.8), Prabhu et al. (1997) (see Figure 2.11) and Livermore and Coronado (2001) (see Figure 2.13) provide some information about the software development life-cycle, the information is limited and does not describe the development life-cycle phases in detail.

In addition, despite the importance of usability engineering processes, the global-SDLC employed to construct software for multicultural groups rarely includes usability evaluation. Even Hall and Hudson (1997), who provide a comprehensive software development life-cycle with internationalisation and localisation activities, do not provide much information about usability issues. They mention in passing the need to achieve usability in their Quality Assurance chapter. Hall and Hudson (1997) also suggest that feedback should be obtained

from the product users throughout the development cycle. However, specific information about including usability activities in the life-cycle is absent. Carey (1998) provided a framework drawn from literature which also lacks a usability evaluation phase. Without a usability focus, the internationalised and localised product may not be usable, and may not be accepted by the target cultural groups.

In sum, while there exists much literature on internationalisation and localisation, there are areas that require attention. For example, much of the work reviewed in this chapter examines only language issues. Deeper cultural elements, which may impact the design of the software, are still neglected.

With regards to the global-SDLC approach to develop software for multicultural groups, there exists information on each phase of the life-cycle. However, little information is available about a complete SDLC process. After all, the software development life-cycle process is not just requirements analysis, design, implementation, usability evaluation, or, internationalisation-localisation in isolation, but an amalgamation of all the phases.

Moreover, while much has been reported on the phases of the SDLC, and internationalisation and localisation, little has been reported about incorporating usability evaluation activities into the software development life-cycle to provide software for multiple cultural groups.

Many authors, such as Nielsen (1990b) and Dray (1996), advise software developers to employ the target cultural groups to evaluate the usability of localised software in the target markets. However, as illustrated by the examples in Section 2.6.4, the efficacy of usability evaluation tools from the developed nations in North America and Europe, may be problematic when employed in nations outside the tools' origin.

2.7 The Gap

Little is known about the usability evaluation phase of the global-SDLC as well as the effectiveness of global-SDLC as a whole. In particular, current usability assessment tools from developed nations, employed as part of global-SDLC to provide software for multiple cultural groups, may not be effective in developing nations.

Chapter 3 Evaluating Usability Assessment Tools

The literature review in the previous chapter uncovered a Gap – usability assessment tools (UATs) from developed nations may not be effective when employed in developing nations. The UATs may not be effective when employed, as part of the global-software development process in evaluating the usability of the software targeted at multiple cultural groups. As defined in Section 2.4, global-SDLC is a software development life-cycle which incorporates the processes of internationalisation and localisation. The next section, Section 3.1, provides the rationale for evaluating the application of the UATs and raises a key question with regards to this evaluation. Section 3.2 presents a possible solution to the key question and Section 3.3 discusses some details of the possible solution. Section 3.4 describes what the work reported in this thesis aims to show.

3.1 Rationale for Examining the Gap

As suggested by the literature review, there is a need to assess the UATs from developed nations employed in the usability evaluation of software targeted at multiple cultural groups. The reasons for investigating this gap are provided in the context of the following areas:

- Localised software and the global-SDLC.
- Usability evaluation of localised software.
- Assessment of imported UATs employed in the usability evaluation phase of the global-SDLC.

The localised software in this section refers to the software that is being evaluated in the usability evaluation phase of the global-SDLC. Recall that localisation is defined as the process of providing the cultural elements of a particular target cultural group. After the software is internationalised, the cultural elements (culture-dependent components) are localised to particular target cultural groups. The localised software comprises the internationalised software using the appropriate message files (utterances), which contain the cultural elements for that

cultural group. It is this localised software that is being evaluated in the usability evaluation phase of the global-SDLC.

3.1.1 Localised Software and the Global-SDLC

Localised software could be provided for societies in developing nations. These societies could use localised software to train and educate their people and thus achieve a computer-literate workforce. Only with a computer literate society can these people take advantage of the benefits provided by ICTs appropriate to their society.

However, many large software companies neglect these developing countries because of the poor returns on the investment which would be necessary to enter the market. This observation is supported by the United Nation's Human Development Report 2001 (UNHDP, 2001) which states, that private sectors that create technology respond to the people that can afford the technologies rather than to those with little purchasing power. Unfortunately, the people in the countries that are neglected are the very people that require localised software for them to apply appropriate ICTs and capitalise on the benefits that they could provide.

Another obstacle to the provision of software in these developing nations relates to the many cultural groups that may exist in any one country. Developing countries usually comprise multiple cultural groups, many cultural groups within one geographical or political area. If the language spoken is used to distinguish the different cultural groups, some countries would have more than 700 cultural groups. For example, Indonesia has 726 living languages (Ethnologue, 2001). A high number of cultural groups suggests that the number of speakers of each language will differ greatly. For example, 26 languages are spoken in Botswana (Ethnologue, 2001) a population of only 1.6 million (FAO/World Bank, 2001). The number of speakers of each language varies. The numbers range from 800 people who speak "/Gwi", a thousand people who speak "//Gana" to about a million people who speak Setswana (Ethnologue, 2001). Given the fact that some languages have a small number of speakers, software companies are unlikely to provide localised/translated software for any cultural group with only 100 or even 1000 speakers. As mentioned in Section 2.6.5, it may be up to the indigenous

population of those countries to provide software for themselves and their fellow nationals.

Given the multitude of cultural groups, this thesis's focus on global-SDLC is relevant and important since global-SDLC is the very process recommended for providing software for different cultural groups referred to in Section 2.4. An important phase in the global-SDLC is the usability evaluation phase. Localised software derived via the global-SDLC must be evaluated by the target cultural groups.

3.1.2 Usability Evaluation of Localised software

Before going into detail about usability evaluation, it is important to remember the formal definition of the term given in ISO/DIS 9241-11 (1998).

> *Usability is "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".*

To users, software products are considered usable if the software is easy to learn and easy to use. Usable *software* actually means usable *user interface*. A user interacts with the system through the user interface. As the entire experience people have with the software is with its interface, if people find the interface usable, then they will deem the entire system usable (Dray, 1995). The focus on developing a usable user interface is crucial to the success of the software.

This focus is also important as the user interface accounts for a considerable portion of the effort in software development. Hix and Hartson (1993) report a study by Myers and Rosson (1992), which states that an average of 48% of a typical application's source code is used to support the user interface in an interactive system. Also, Dray (1995) reports that the user interface account up to 60% of the source code, and a third of the software development project budget.

In addition, there are many benefits associated with usable software. Donahue (2001), Landauer (1995), Dray (1995) and Bias and Mayhew (1994) highlight these including a reduction in training and support costs, reduction in software development time, increased revenues, increased marketability, reduced user errors, and reduced maintenance costs (Karat, 1994). For example, Karat

(1994) reports returns, which ranged from $2 to $500, for every $1 invested in usability related activities in the improvement of the product.

Thus, the user interface is as important to the success of the software as the functionality of the computer system (Hix and Hartson, 1993). Specifically, a *usable* user interface may determine the success of the software, that is meeting the needs of the target cultural groups.

For the localised software to be a success, there is therefore a need to ensure the localised software is usable. Usability evaluation is one way of improving and determining the success of the localised software. Usability evaluation is concerned with collecting data about the usability of a product (or design) by a specific group of users for a specific activity within a particular environment or work context (Preece et al., 1994). In the software development process, usability evaluation activities can be conducted throughout the software development life-cycle, such as requirements analysis and design phases, not necessarily only in the usability evaluation phase. The data collected during the evaluation can then be used to improve the software; furthermore, it can also be used to determine the usability of the product.

Since the localisation phase is usually only conducted towards the end of the software development life-cycle, after the internationalised program design/functions has stabilised, the localised software application is close to the final product that will be released to the target market. Usability evaluation at this stage can thus indicate the likely success of the software; it determines whether the software will meet the users' need as well as determining the usability of the localised versions of the software. Also, the usability evaluation indicates success (or otherwise) of global-SDLC processes employed to develop the localised software. If the software development process has been successful, software that meets the users' need would be obtained.

In the usability evaluation of the localised software, authors such as Nielsen (1990b) and Dray (1996) recommend that members of the target cultural groups of the localised software participate in the usability evaluation. They also suggest that the usability evaluation be conducted in the target cultural group's environment. These users are representative members of the people who will use the localised software. The employment of representative users is important, since

they would be able to identify elements in the user interface or functionality which are inappropriate for them. For example, only indigenous users would be able to identify problems with language translation, or inappropriately localised icons. Although many authors, such as Dray (1996), Russo and Boor (1993), Nielsen (1990b), supply information on usability evaluation of adapted software in the target markets, little has been reported about effectiveness of these extant UATs when employed in target markets such as developing countries. This lack of research may be due to the lack of interest in providing software for developing countries. Furthermore, some of the information available on UATs employed internationally, such as Modica and Fiedler (1999), Herman (1996) and Fernandes (1995), are based on authors' experience rather than on an evaluation of the UATs as detailed in Section 2.6.4. The information provided by anecdotal evidence may not be accurate in the first place, and needs to be validated by more rigorous studies.

In sum, usability evaluation is important in the development of localised software. However, there is a lack of information on the effectiveness of extant UATs employed in the usability evaluation phase. In addition, details about the employment of UATs in countries outside North America and developed European nations may not be accurate given their anecdotal origins. As such, if UATs in the usability evaluation phase do not work (perform as expected), software that does not meet requirements or software with usability problems may be inadvertently released. Thus, there is a need to evaluate the UATs in various cultural contexts to determine the effectiveness of UATs in the target cultural groups.

3.1.3 Assessment of UATs Employed in Usability Evaluation

This need to evaluate UATs stems from the fact that imported UATs may not work when employed in the target cultural group's environment. UATs are deemed to be imported tools since nations, such as developing nations, typically do not have their own UATs, employing instead existing tools imported from developed nations. To ascertain if the imported UATs are effective in the new environment, there is a need to evaluate them.

Lack of evaluation of UATs is similar to the lack of evaluation of the software development approach. Fenton, Pfleeger and Glass (1994) maintain that

the effectiveness of new practices (proposed by software researchers to improve software development and maintenance) is rarely, if ever, supported by hard evidence. Robinson (2001; p111) refers to software engineering as a practice-oriented discipline. He reports "We simply don't have enough information about the actuality of practice to be certain that our research efforts are addressing the significant problems of a practice-oriented discipline." (Robinson, 2001; p111). Glass (1994; p.44) further notes that almost no computing research had an evaluative phase.

According to Glass, the evaluative phase is where the research evaluates a proposal or analytic finding by conducting experiments, or by observing (case study or protocol analysis). The evaluation may lead to a validated model, principle, or theory (Glass, 1994; p.44). Furthermore, Glass expresses his surprise that "establishing pilot studies to test out ideas in a realistic setting and evaluate their success, incredibly enough, was not done" (Glass, 1994; p.45). Glass also advises researchers that the only way to determine whether new ideas are of any value is to test them in a practical setting. For results to be of value, the experiment should be conducted in a realistic setting and not in "toy situations". Fenton, Pfleeger and Glass (1994) define "toy situations" as contexts involving "artificial problems in artificial situations".

Thus, before artefacts – theories, guidelines, methods, tools, techniques – can be useful, researchers must evaluate these artefacts to ensure they work in the various contexts targeted. In the case of imported UATs, if the imported tools are applied in a different cultural context, that is, a different target market, the tools should be evaluated in that target market to ensure they work as expected. Only through evaluating the UATs in a realistic setting, can software engineers:

- Identify which UATs employed in the global-SDLC are effective
- Determine why certain UATs work or do not work in a real setting
- Ascertain in what contexts the UATs are effective

In sum, the reasons and context in which usability evaluation of localised software is focussed, are as follows:

Firstly, there is a need of localised software especially for those nations which need ICTs to improve their livelihood. As those nations usually have many cultural groups, global-SDLC is recommended since it is employed to provide software for multiple cultural groups.

Secondly, localised software must be usable at least equally to or more usable than the un-localised version. To determine and improve the software's usability, usability evaluation is employed. Sound processes, be it in usability evaluation or the global-SDLC, are needed to provide successful localised software.

Thirdly, given the suspicion that imported UATs may not work in the target markets, an evaluation of the UATs should be conducted. There is a need to determine if imported UATs will be effective in reaching their goal, that is, determining the success of the localised software. The following key question is therefore raised.

Key Question:	*Are imported usability assessment tools appropriate for the usability evaluation – as part of the global-software development life-cycle – of internationalised and localised software in all cultural contexts?*

3.2 A Possible Solution

A possible answer to the Key Question is to apply imported UATs in a realistic setting, that is, apply the UATs to evaluate the usability of a localised software application.

Theoretically, if software developers follow the prescribed guidelines in evaluating the usability of the localised software, the evaluation results should be able to provide an indication whether the localised software was successful. The success of the software would also indicate the success of the global-SDLC processes, since successful software is derived from the successful implementation of the global-SDLC process.

Traditional SDLC	Global-SDLC Activities	I18n	I10n	UE
Requirements Specifications	• Identify target cultural groups of software	✓		
	• Gather information about target cultural groups	✓	✓	✓
	• Identify cultural elements that impacts design and architecture of internationalised product	✓	✓	✓
	User Interface Design	✓	✓	✓
	• Design–Global Design			
	• Build–Prototype localised versions			
	• User testing–Usability evaluation/formative evaluation employ natives to evaluate the prototypes			✓
	• Iterate the cycle design-build-test			
Systems Design	• Design generic core: identify which cultural elements to include or exclude	✓		
	• Define module components and relationship between the modules			
Implementation	• Write the source-code of the generic core			
	• Localise the culture-dependent elements for the target group. For example, translating messages and provide cultural elements, such as icons		✓	
Testing	• Ensure all functions work according to specifications			
	• Ascertain performance of software is acceptable			
	• Determine if the localised message files work with the generic core			
Usability Evaluation	• Summative evaluation–Evaluate the usability of localised version of software			✓
	• Employ natives and intended users in the usability evaluation			
	• Identify problems to fix in the localised software before release			

Table 3.1: Simplified steps of Global-Software Development Life-cycle
The ticks ✓ indicate the steps conducted by the internationalisation team (i18n), localisation team (l10n) or usability evaluation team (UE). These teams are the people involved in the global-software development.

3.3 Details of the Possible Solution

Even though the focus of this thesis is on the evaluation of the usability tools, the assessment of the UATs in the usability evaluation phase cannot be studied in isolation; the evaluation of tools must be studied within a specific context.

As the product being evaluated is a localised software application, the localised software must be first developed. The recommended process in which the localised software is developed is through the process of global-SDLC. The global-SDLC is assembled based on current internationalisation and localisation practices. Table 3.1 contains a summary of the various steps of the global-SDLC.

Once the localised software is derived, the usability assessment of the localised software can take place. Every UAT has distinct strengths and inherent weaknesses (Doubleday et. al., 1997; Henderson et al., 1995; Jeffries et. al, 1991; Karat, Campbell and Fiegel, 1992; Yamagishi and Azuma, 1987). To offset the individual weaknesses of each UAT, a more comprehensive usability evaluation could be obtained by using multiple UATs. This method is known as triangulation or consensual validation (Holleran, 1991). Rather than relying on a particular UAT (Doubleday et al., 1997; Holleran, 1991), multiple UATs are employed to record the same phenomenon.

Data collected from the multiple tools are compared. The comparison is to determine if the data from the different UATs are consistent in indicating the success or failure of the localised software. If any of the UATs fail, assuming not all of them fail, the inconsistent data would alert the experimenter to the failure. Further investigation should then be conducted to determine why the UATs failed, and identify under what circumstances they had failed.

Nielsen (1990b) suggested that the usability of localised software should be evaluated by local target users in the target market. A local usability engineer could be employed to conduct the usability evaluation studies (Nielsen, 1996). By employing a local person, the problems associated with using an interpreter in a non-English-speaking target market may be avoided. Problems, such as details or nuances missed in the translation by an interpreter, may be prevented.

3.4 What was Attempted in the Study

The aim of this thesis is to determine if imported UATs can be employed in the usability evaluation of localised software developed using the global-SDLC. To answer the key question, UATs were employed in the usability evaluation of a localised software application, *Hamparan*.

The framework applied in this thesis is depicted in Figure 3.1. The key question raised is depicted at the start of the process (round-cornered box at the top of Figure 3.1). After the start of the process, an existing spreadsheet was internationalised and localised via the global-SDLC process. To ensure that the guidelines were adhered to, checks were conducted at various points of the global-SDLC. For example, after the implementation stage, testing was conducted to determine if set criteria, like "Were all the menus translated?", were met. The shaded box in Figure 3.1 indicates the focus of this thesis. To determine the appropriateness of the UATs, the data collected from the UATs were evaluated. The steps carried out to collect the data are detailed in Chapter 4.

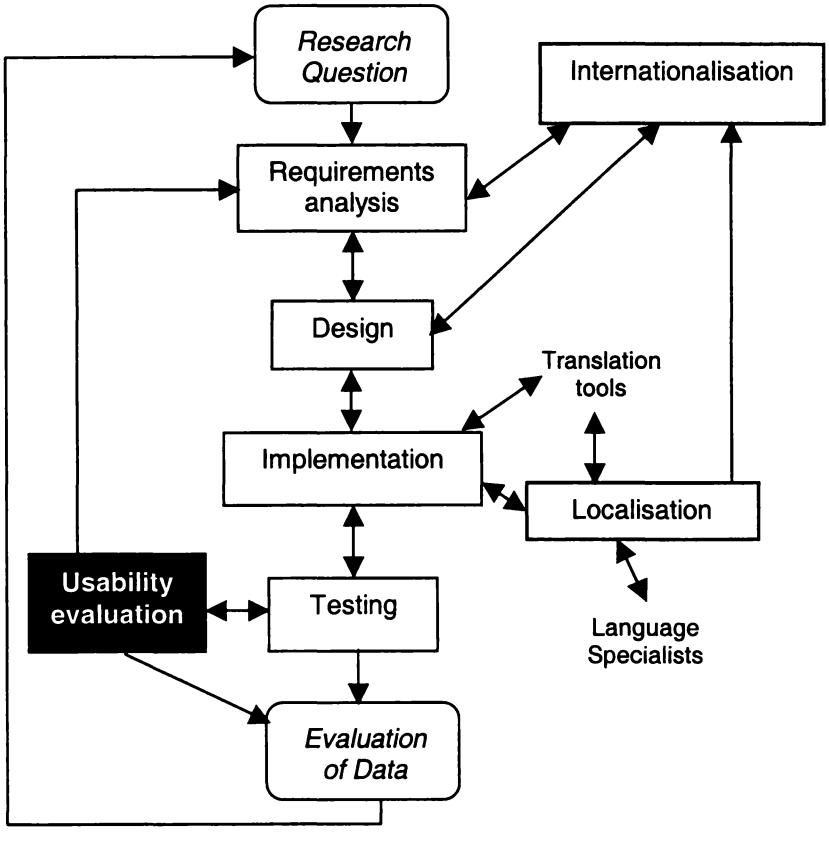

Figure 3.1: The Global-SDLC Model – adapted from Kano (1995)
The rectangles are phases of the global-SDLC whereas the round-cornered rectangles are components of this study.

Chapter 4 Method

The key question posed in Chapter 3 was whether imported usability assessment tools (UATs) are appropriate for the usability evaluation of internationalised and localised software in all cultural contexts. To answer this key question, data was collected using imported UATs applied in the usability evaluation of a localised software application. The data collected was then examined to determine the appropriateness of the UATs in the usability evaluation process. However, before the usability evaluation could take place, the localised application had to be developed first. Section 4.1 provides details of the application of the global-SDLC process to the development of an internationalised and localised software application. Section 4.2 describes the usability evaluation of the localised software and describes how the data was employed to determine the appropriateness of the UATs in the usability evaluation.

4.1 Adaptation of the Software Application

This section describes a software application adapted for use by different cultural groups. The software was adapted using the global-SDLC. The software application identified for the adaptation process was TCALC (Borland International Copyright). TCALC is a US English spreadsheet that was bundled with Borland's Turbo C++ compiler. The spreadsheet was selected since it had the basic functionalities of existing commercial spreadsheet programs. Moreover, TCALC's source code was available.

Although the process of internationalisation and localisation of software should address cultural factors deeper than those at the symbols layer (practised by software developers today), TCALC was adapted at the language level only. This simplistic adaptation does not affect the focus of the thesis. The thesis focus is the evaluation of the effectiveness of the UATs employed as part of the global-software development process. In other words, for this present work, it is important that a localised product is developed using global-SDLC guidelines – including some form of internationalisation and localisation.

4.1.1 Requirements Analysis

As depicted in Table 3.1, at the beginning of the global-SDLC process, software engineers have to identify the target cultural groups of the software application. They should gather information about the target cultural groups, for example identify which cultural elements will impact on the design of the internationalised product. In this thesis, the target cultural group focussed on was Bahasa Melayu-literate Malaysians. Bahasa Melayu is Malaysia's national language and is spoken and used by almost all Malaysians. The Malaysian cultural group was chosen, as the localiser, the author of this thesis, is himself a Malaysian. Hence, the background knowledge to internationalise and localise the software application for Malaysian users, was available.

Furthermore, representative users required to evaluate the usability of the localised software, were readily available at a Malaysian university. Besides the Bahasa Melayu speakers, the spreadsheet was also targeted at Iban speakers. Iban is the language of the indigenous Iban ethnic group in Malaysia. In this exploratory work, only the usability of the Bahasa Melayu version was investigated.

User Interface Design (Design-Build-User Testing Iterate-Cycle)

The next step of the global-SDLC is the user interface design. Since the functionality of the spreadsheet was not altered; only the language of the user interface was modified. The main interest of this study was to ensure that members of the different target cultural groups were able to interact with the spreadsheet. A Bahasa Melayu prototype was developed to test whether the localisation was achievable. The visible items on the interface, such as menu names, were translated to Bahasa Melayu. The author of this thesis, who is literate in Bahasa Melayu, conducted this translation. This translation involved substituting the English messages (in TCALC's source code) with Bahasa Melayu messages. No other changes were made. The source code was compiled and run. This prototype showed that TCALC could be localised to accommodate Bahasa Melayu.

4.1.2 Design of Generic Core

According to the global-SDLC process described in Section 3.3, the goal in this phase was to acquire a generic core design by internationalisation.

In this phase, the author identified and extracted the culture-dependent/language-sensitive components from the source-code. The language-sensitive items included error messages, dialogue messages, menu items, prompts, instructions, currency symbol, and status bar displays of cell content, such as text, value, or formula. Most of these language-sensitive messages were specified as pre-processor directives in TCALC's source code. For example, the error message MsgNoOpen in TCALC's source code was written as:

```
# define MsgNoOpen "Cannot open the file."
```

On compilation of the file with the above statement, the pre-processor changes all occurrences of the identifier MsgNoOpen in the source code to "Cannot open the file.". Thus, the entire TCALC's source code was searched for such identifiers. All the culture-sensitive messages specified as "#define" statements were identified and placed in a separate file, that is, a message file. With this modification, a base generic core without the culture-dependent components, was obtained.

In this adaptation, only one module was designed. This new module was designed such that the generic core could use the message files of languages with Latin character sets. This module provided functionalities that would be able to load all the different messages and switch languages, that is, using different message files at run-time. To ensure that all the language-sensitive components from the source code were identified, a user guide for TCALC was written up. This user guide (see Appendix A) was used later as a check-list to make sure that all the items that needed to be localised, were localised.

4.1.3 Implementation

In this phase of the global-SDLC, the generic core was implemented. The internationalised version of TCALC is called First Internationalised Research Spreadsheet Tool (FIRST). The software was implemented in such a way that users can switch to different languages at run time.

All the culture-dependent pre-processor directives in TCALC's source code were replaced as variables. By making this change, each variable at run-time would contain the same message in meaning but in a different language. A function, LoadMsgDb(language), was added to FIRST. This function loaded all the messages in the message file into the corresponding message variables. For

example, the variable MsgOverWrite will contain the string "The file exists. Do you want to overwrite it?" when the English version of FIRST is used. When Bahasa Melayu is selected, the variable will contain the string "Fail tersebut wujud. Mahukah anda menulisgantikan fail lama?". Messages in the different target languages are shown in Appendix B.

To determine which language was used at run-time, the user was prompted to choose a language from a list of languages. This list of languages was displayed on the screen when FIRST was run (see Figure 4.1). Once the user had selected the language, other languages could be selected using the ALT-key combination. Pressing ALT-B at run-time will load the Bahasa Melayu version, ALT-E the English version, and ALT-M the Māori version. Figure 4.2 shows the language table, LangList.txt, which lists the message files to be loaded and the designated ALT key to be used. When ALT-B is pressed, the messages in the message file "bmelayu.msg" are loaded into the message variables in FIRST. The Bahasa Melayu version is called *Hamparan* (meaning spreadsheet in Bahasa Melayu), the Māori version is known as *Te Ripanga* (meaning spreadsheet in Māori), and the Iban version, *Pengancau* (meaning "something that can be spread" in Iban).

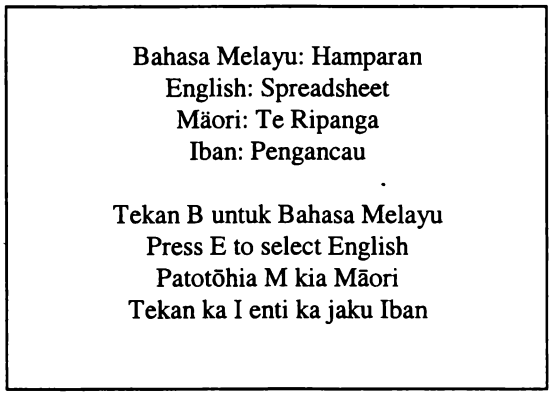

Figure 4.1: Prompt screen with four languages

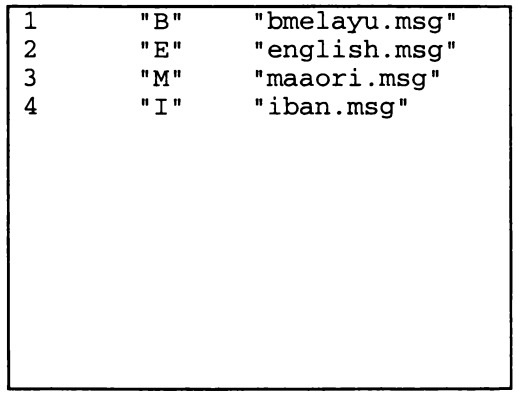

Figure 4.2: Language table with four message filenames

The outcome of the implementation phase shows that the software has been internationalised to accommodate three new languages. Figure 4.3 and Figure 4.4 show the screen of FIRST (after the implementation) in English and in Bahasa Melayu respectively. The localisation of the messages is detailed in the next section.

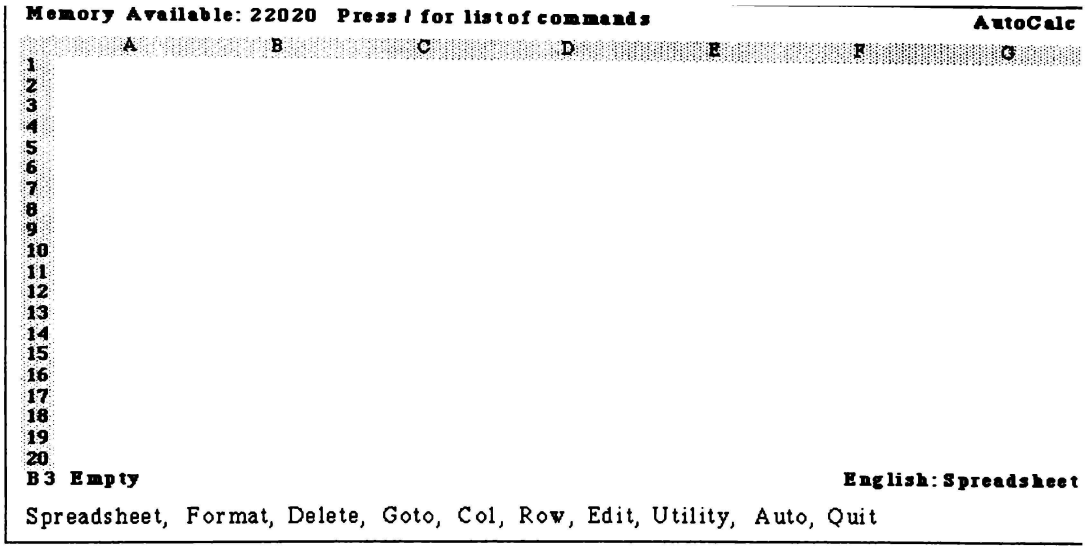

Figure 4.3: Screen dump of English interface

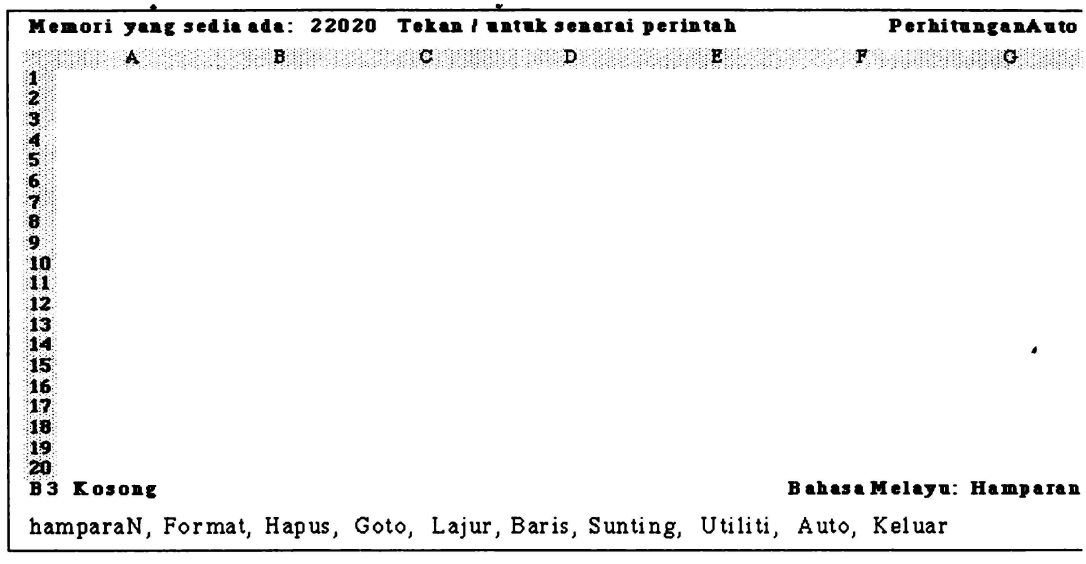

Figure 4.4: Screen dump of *Hamparan*'s Bahasa Melayu interface

4.1.4 Localisation

While the modification of the generic core was conducted, the localisation of the message files was carried out. The initial translation of messages carried out by the author in the user interface design, in Section 4.1.1, was not used. Instead, the English messages in the message file were translated to Bahasa Melayu by a Malaysian undergraduate who was majoring in linguistics and Malay studies at a Malaysian university. The translator was fluent in Bahasa Melayu and was computer literate. She was able to translate the messages in the context of computing. Translators who were not familiar with computing terminology may make mistakes such as those reported by Russo and Boor (1993). One such error involved the term "menu" being mis-translated into "a list of food items" (Russo

and Boor, 1993). This English to Bahasa Melayu translation was conducted with reference to the *Istilah Komputer* (Dewan Bahasa dan Pustaka, 1993) which contains a glossary of computer terms in Bahasa Melayu, and a Lotus 1-2-3 tutorial reference book in Bahasa Melayu (Yong, 1995).

To ensure the initial translation was accurate, Uren, Howard and Perinotti (1993) suggest conducting a back-translation. Thus, a second translator (the author) back-translated the Bahasa Melayu messages to English. If there were any discrepancies between the original English and the back-translated English version, corrections were made to the final Bahasa Melayu messages.

During the back-translation process, the author also collaborated with other native Bahasa Melayu speakers. This collaboration process was carried out as a precaution, in case the principal translators were influenced by their contact with English software. Detailed information on this translation technique has been reported in Yeo and Barbour (1997). The translation of the English messages into Māori is detailed in Barbour (1996), whereas the translation of the English messages into Iban is reported in Yeo and Barbour (1997).

4.1.5 Testing

Once the generic core is implemented and the localisation step is completed, all the localised versions should be tested. In the testing phase, the goals are to:

- ensure all the functions of the localised versions work
- ascertain the performance of software is not compromised, and
- ensure that the localised components work with the generic core (McHugh, Honkela, and Hudson, 1997)

In this study, only *Hamparan* was tested. All the functions detailed in the TCALC user guide (see Appendix A) were tested to determine whether *Hamparan* version operated as well as TCALC.

As a result of this testing, an overlooked culture-sensitive component was identified. This culture-sensitive component was the Yes-No response to certain prompts and questions. The component was overlooked as it was "buried" in the code unlike the messages which were written as "#define" statements. The Yes-No component was extracted from the source code internationalisation step and

replaced with a variable. "Yes-No" was then translated to "Ya/Tidak" – the localisation step. The process of resolving issues related to the Yes-No component is an example of iteration that occurs between the implementation and testing phases.

The performance of the localised version was also examined to ascertain that there was no noticeable decrease in speed in the execution of the software. All the interactions described in the user guide were carried out in *Hamparan*. The Māori and Iban versions were tested only to the extent that FIRST could switch to these languages during run-time. This test was conducted to ensure that the spreadsheet was internationalised, such that it could accommodate the target languages of Bahasa Melayu, Māori and Iban.

This phase also provided a check to determine if the internationalisation process was done effectively. From a technical point of view, the localised messages worked with the generic core. The implementation and testing of FIRST has been reported in Barbour and Yeo (1996). After the implementation and testing of FIRST, the next phase is the usability evaluation.

4.2 Usability Evaluation of *Hamparan*

The internationalised and localised spreadsheet FIRST was produced according to global-SDLC guidelines. It allows users to interact in English, Bahasa Melayu, Māori and Iban. This section provides details of the usability evaluation, the next phase of the global-SDLC, of *Hamparan*. As the focus of the research is on the appropriateness of UATs, details of the assessment of imported UATs employed in the usability evaluation phase, are provided in Section 4.2.1. Section 4.2.2 describes the specific UATs employed in the usability evaluation. Sections 4.3 onwards provide information about the materials and methods employed to assess the UATs' suitability in the usability evaluation phase.

4.2.1 Experiment Design

It will be remembered from Chapter 3 that the key question is: Are imported UATs appropriate for the usability evaluation, as part of the global-SDLC, of internationalised and localised software in all cultural contexts? The experiment described in the following section was conducted to collect data in an attempt to provide an answer to the key question.

The usability evaluation was conducted in one cultural context, in Malaysia. A Malaysian experimenter (the author) carried out the usability evaluation using representative users of *Hamparan*. The representative users were Malaysian Bahasa Melayu speakers. This context is in accordance with recommendations made by Nielsen (1990b) who suggested employing representative users from the target cultural group. Nielsen (1996) suggested conducting the usability evaluation in the target user's premises as well as employing a local usability consultant to run the evaluation.

As mentioned in Chapter 3, the data collected from the usability evaluation can be employed to achieve two goals.

a. Data collected can be used to identify the changes to make to the interaction design and improve the design's usability (Hix and Hartson, 1993). This is called formative evaluation (Hix and Hartson, 1993).

b. Data collected can be applied to determine the effectiveness of the product (Wright and Monk, 1991). This is called summative evaluation.

In order to ascertain whether the imported UATs employed, were appropriate in the usability evaluation, two aims in relation to the above evaluation goals were used. The imported UATs would be deemed appropriate for the usability evaluation of *Hamparan* in the described Malaysian context if the data collected while using the imported UATs were able to contribute the following processes:

a. Data collected using the imported UATs can be employed to improve the usability of *Hamparan* (a formative-evaluation)

b. Data collected using the imported UATs can be employed to determine the usability of *Hamparan* (a summative-evaluation)

Although formative evaluation is normally conducted in earlier stages of the life-cycle (Hix and Hartson, 1993), UATs in this study were employed to collect data near the end of the life-cycle process. Formative evaluation of *Hamparan* conducted at this stage was appropriate since the goal was still the same; the data collected was to be employed to improve the software – albeit in

this case, improving the next version of the software. Given that the collected data is applied to improve the spreadsheet, usability data collected by the UATs for the formative-aim should be qualitative data, such as, a list of problems that users had experienced while using the system being evaluated (Hix and Hartson, 1993).

With regards to summative evaluation, the evaluation is usually conducted at the end of the development process, for example, when the interface or system is complete, or near complete (Hix and Hartson, 1993). In this experiment, *Hamparan* was considered a "near-complete" application. Summative evaluation of *Hamparan* conducted at this stage of life-cycle was thus appropriate.

Wright and Monk (1991) state that summative evaluation provides summary data on effectiveness of a system. In this experiment, the data collected during the summative-evaluation is used to assess the usability of the software: the types of data collected could be performance measures or subjective measures. Performance measures consist of counts of actions and behaviours that can be observed, while subjective measures pertain to people's perceptions, opinions, and judgments (Dumas and Redish, 1993).

At this point, it is appropriate to point out that while the data collected can be used to identify the areas of the interface which need to be improved, the data collected do not indicate how the improvements of the usability of the interface are to be carried out. The data collected in this study was not used to improve the usability of the spreadsheet, but applied to determine the efficacy of the UATs.

After the goals of the usability evaluation were defined and the type of data required to achieve the goals were identified, the next step was to select the UATs to collect the necessary data.

4.2.2 Usability Assessment Tools

To assess whether the UATs were appropriate for the usability evaluation, more than one UAT was employed in the experiment. As reported in Chapter 3, every UAT has inherent strengths and weaknesses. Triangulation of UATs was carried out, that is, multiple UATs were employed to take advantage of each UAT's strengths. The use of multiple UATs would also provide greater confidence in the data collected if the data collected from the multiple UATs were in agreement. Thus, the following imported UATs were selected: logging-

augmented think aloud, SUS (a questionnaire), and interview. These UATs are considered imported tools since these tools originate from overseas and were employed in Malaysia, a developing nation. These UATs were also chosen as they are commonly referred to in the literature of usability evaluation (Henderson, et al., 1995).

Logging-augmented Think aloud

The first UAT selected was think aloud. The think aloud method requires the participants to verbalise their thought processes while they complete assigned tasks. When used in this manner, the think aloud method, provides a wealth of data which can be analysed both qualitatively and quantitatively (Preece et al., 1994). Nielsen (1993) and Yamagishi and Azuma (1987) reported that think aloud method could be used to isolate specific usability problems. Performance measures can also be obtained from think aloud data, for example, the counts of observed behaviours. The think aloud method is one of the most-used UATs in industry (John and Marks, 1997), which is perhaps why Nielsen (1993) claimed that it "may be the single most valuable usability engineering method". However, some authors have voiced their reservations about the think aloud method. One argument against the use of think aloud is that a person who is thinking aloud would detrimentally affect the user's performance in completing the tasks. However, Hix and Hartson (1993, p.206) noted that think aloud evaluation does not measurably affect task performance, except for low level tasks that occur in a very short time (a few seconds). Preece et al. (1994) commented further that some participants are uncomfortable in thinking out loud and that participants may lapse into silence. Participants who lapsed into silence can be prompted to talk. Participants who are uncomfortable about thinking aloud, however, may continually lapse into silence. Thus, the think aloud method was augmented with a logging mechanism that recorded all the keystrokes made by the participants while using the software. The logged data was used as a record for later examination of what the participant was doing if he or she lapsed into silence during the think aloud session.

System Usability Scale

The second UAT selected was a survey method, that is, the System Usability Scale (SUS). The SUS is a Likert-scale questionnaire which provides a global view of the participants' subjective evaluation of a product's usability

(Brooke, 1996). A Likert-scale questionnaire contains questions which participants indicate their agreement or disagreement along a five-point (or sometimes longer) scale ranging from "strongly agree" to "strongly disagree" (Burns, 1994). The SUS is a reliable method to gauge a product's usability: a University of Cork study placed System Usability Scale's correlation reliability of 0.8588, on par with the Systems Usability Measurement Inventory (SUMI) (Holyer, 1994). SUMI is a commercially available questionnaire which is also used to evaluate systems (Porteous, Kirakowski, and Corbett, 1993). In this study, the SUS was also chosen because it was easy to both administer and score. The short time needed to complete the SUS was important, as the participants in the experiment were university staff with limited time available.

Interview

The third UAT employed is another survey tool, the interview. The interview technique is typically used to obtain the users' opinions, preferences, impressions and attitudes about the product being evaluated (Dix et al., 1998; Preece et al., 1994). Usability problems can also be identified from the interview data (Henderson et al., 1995; Yamagishi and Azuma, 1987). In addition, Yamagishi and Azuma (1987) contended that usability problems isolated from interview responses were more general, as compared to usability problems identified from think aloud data, which were more specific.

4.2.3 Comparison of Data Collected

To determine whether the UATs were appropriate for the usability evaluation of *Hamparan*, the data collected using the UATs were applied in the following manner. Qualitative data in the form of usability problems was obtained from the think aloud and the interview data. The formative-process would be successful if the usability problems identified from both think aloud and interview were sufficiently detailed. Sufficiently detailed means that the information provided was detailed enough for a software engineer to suggest a possible solution to fix the problem. To ensure data collected were reliable, the data collected from the augmented think aloud and interview were also compared. As Yamagishi and Azuma (1987) state, the data collected from the think aloud and interviews were "mutually consistent"; the data collected from the think aloud and interview in this study would be able to cross-validate one another.

For the summative-process, data from two types of measures were obtained, performance and subjective measures. The performance measures were obtained from the think aloud data. These performance measures were counts of positive and negative behaviour observed in the think aloud session. Positive behaviour includes participants making a positive comment about the spreadsheet, whereas the negative behaviour includes participants expressing their frustration, or making a negative remark. The positive and negative behaviours were regarded as reliable measures given that these behaviours were immediate; attempts to alter and falsify these behaviours during the think aloud session would be detectable. For example, it would be quite obvious and out-of-place if participants expressed a positive comment when they were struggling to complete a task. Furthermore, these behaviours reflect the performance of the participants while using the spreadsheet. If the participants had relatively more positive than negative behaviours, then the performance of these participants are expected to be better than participants who had fewer positive compared to negative behaviours.

The other type of data, subjective measures, was obtained from the SUS and interview. The SUS scores provide an indication of the participant's opinion of the product's usability. In the interview, the participant's opinions of the product were identified from their interview responses.

To determine if the summative-process was successful, data from all these three sources were compared. The three data measures were expected to agree. Only when the data are consistent can the data give an indication of the software's usability. The assumption here is that a participant who performed well during the think aloud (that is, few negative behaviours or many positive behaviours observed in the think aloud data), would give favourable subjective scores in the SUS ratings as well as the interview comments. This premise is based on Nielsen and Levy's (1994) study. Nielsen and Levy (1994) indicated that performance and subjective/preference measures positively correlate; good performance such as fast task completion time indicated by the performance measures would indicate positive ratings in subjective data such as preference ratings. This correlation was discovered from a meta-analysis of 113 pairs of objective data (errors rate and/or time taken to complete tasks) and subjective data (preference ratings) drawn from 57 studies reported in leading human-computer interaction conferences and journals (Nielsen and Levy, 1994).

In sum, if the data collected using the imported UATs achieved the two aims, the imported UATs would be deemed appropriate when employed in the usability evaluation of *Hamparan* (localised software developed from the application of the global-SDLC) in Malaysia (a specific cultural context).

The descriptions of materials and procedures used in the experiment are provided in Sections 4.3 to 4.8. Details of the data collection and the data analysis procedures are reported in Section 4.9.

4.3 Pilot Studies

Before the main experiment was carried out, two pilot studies were conducted. The first pilot study group comprised 11 participants who were graduate students of a New Zealand university. The participants consisted of nine graduate students with experience in using spreadsheets and two graduate students without experience in using spreadsheets. This study was conducted to establish a set of tasks that could be assigned to the participants in the main experiment. A list of tasks that could be completed on FIRST (with the English interface) was assigned to the participants. The assigned tasks mirrored tasks typically done in spreadsheets, for example, creating a formula, and saving a spreadsheet. The participants were expected to complete the tasks in 30 minutes. The final list of tasks used in the main study provided in Section 4.5 was very similar to the initial list of tasks used in this pilot study.

A second study was conducted to test the tools and procedures in Malaysia. In this second pilot study, seven participants used *Hamparan* to complete the set of tasks obtained from the first pilot study. All participants were staff members of a university in Malaysia. The participants' experience ranged from novice spreadsheet users (those who had used a spreadsheet less than five times), to experienced spreadsheet users (those who had used spreadsheets daily). This second pilot study provided an opportunity for the experimenter (the author) to test the equipment and procedures as well as the UATs to be used in the main experiment.

Hours of spreadsheet use per week	No. of participants
<1 – 4	13
>4	4
Total	17

Table 4.1: Hours of spreadsheet use per week

Total hours of spreadsheet use in whole computing "career"	No. of participants
< = 40 hours	4
> 40 hours	13
Total	17

Table 4.2: Total number of hours of spreadsheet experience

Occupations	Status	No. of participants
Manager	Highest	6
Lecturer	High	2
Systems Analyst	High	1
Tutor	High	1
Assistant Accountant	Low	1
Administrative Assistant	Low	1
Data Processing Operator	Low	1
Clerk	Low	3
Typist	Lowest	1
Total		17

Table 4.3: Occupation of participants
The order of the occupations depicts the seniority in the organisation.

4.4 Participants

Seventeen spreadsheet users were recruited to participate in the main experiment. These participants were representative of users who use spreadsheets for their work. They are also the people who will use any new software, such as software with Bahasa Melayu interfaces, introduced to the workplace by their employers. These users were all staff members of a Malaysian university and had used spreadsheets in their work. Their reported use of spreadsheets ranged from at least an hour a week, to four hours a day. Participants who were computer literate and had used spreadsheets were recruited, as it was believed that they would be able to transfer their knowledge of one spreadsheet to another. These characteristics were preferred as the participants would be able to explore the spreadsheets with little or no training. If training were provided, the time the participants would be away from work would be greater. As all experiments were

conducted during the office hours, the time for running the whole experiment was kept to a maximum time of one hour for each participant.

Available participants from different levels of the organisation were recruited. The participants' occupations ranged from clerks to managers. This diversity was to allow for users' perspective from different levels of the organisation as spreadsheet users are found at all levels in the university. Summaries of the participants' profiles are provided in Table 4.1, Table 4.2, and Table 4.3. In Table 4.3, a person is considered of high status if he or she has a Bachelor or higher degree. A more detailed profile of all participants is available in Appendix D. All participants were at least bilingual, that is, able to speak both Bahasa Melayu and English. However, the experimenter did not request the participants speak only English, or only Bahasa Melayu, in the experiment. In addition, given the exploratory nature of this study, the influence of language used by the participants to communicate with the experimenter was not examined.

1. Enter all the data (see Table below) into the Spreadsheet. *Masukkan data ke dalam Hamparan.*

	JANUARY	FEBRUARY	MARCH
Rent	200.00	200.00	200.00
Food	230.00	340.00	315.00
Electricity	55.00	56.00	45.00
TOTAL	485.00	596.00	560.00

2. Use formulae for the TOTALs. *Gunakan rumus untuk TOTAL.*

3. Save your spreadsheet. *Simpankan hamparan anda.*

4. Increase the column width to fit the text "Electricity". *Besarkan lebar lajur untuk mengandungi teks "Electricity"*

5. Change the TOTALs to currency format. *Tukarkan TOTAL ke format wang RM.*

6. Quit from the Spreadsheet. *Keluar dari Hamparan*

Figure 4.5: Spreadsheet tasks assigned to participants

4.5 Tasks

Six common spreadsheet tasks were put before the participants in the think aloud session. These six tasks were used in order to get participants to explore the various functions of the spreadsheet. The first five tasks were arranged according to increasing difficulty. More difficult tasks required more

steps to complete. The first and last tasks were the easiest. Figure 4.5 below shows the six tasks as presented in the instruction sheet given to the participants.

4.6 Experiment Environment

The experiment was conducted in an office at a Malaysian university, where all the participants worked. The participants used *Hamparan*, which was run on a personal computer. Three UATs were employed to collect data. During each participant's session, the experimenter was also present in the office. Figure 4.6 depicts the room layout.

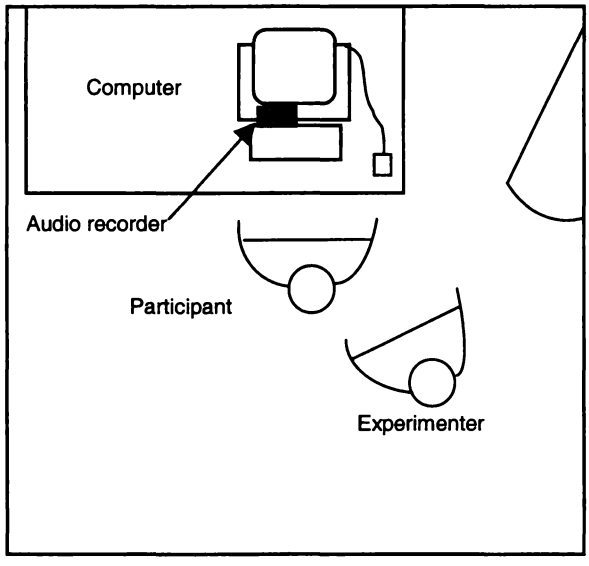

Figure 4.6: Room layout for experiment

4.7 Experimental Protocol

The participants were recruited through a phone conversation and an appointment for their participation was made. On the arrival of the participant at the appointed time, the experimenter introduced himself and gave a brief description of the study. The experimenter then elicited information about the participant's spreadsheet experience. All the participants' responses are provided in Appendix D. Before participants began the experiments, they filled in a consent-to-participate form (see Appendix C.1).

Next, participants were asked to complete the set of tasks described in Section 4.5 using *Hamparan*. The participants were then requested to think aloud while they completed their tasks. After the think aloud session, the participants were asked to evaluate the spreadsheet's usability by completing the System Usability Scale (SUS). Lastly, the participants were interviewed to obtain their

opinions concerning the spreadsheet they had just used. They were also asked to give suggestions on how to improve *Hamparan*.

After the interview, a debriefing was conducted. During this debriefing, the experimenter answered all the questions the participants had about *Hamparan*. If the participants did not complete any of the tasks, the experimenter showed the participants how to complete those tasks. An effort was made to be consistent in conducting the experiment. For example, instructions were read from a pre-prepared script, see Appendix C.2. A checklist was used by experimenter to ensure that all participants completed the same activities. All usability evaluation sessions were conducted by the same person, the author.

4.8 Data Collection

The following sections detail the procedures employed to collect the usability data. The sections are provided in the order in which each UAT was used in each participant's session, that is, logging-augmented think aloud, SUS and interview.

4.8.1 Think Aloud

The participants were required to think aloud, that is, provide an on-going commentary of what they were doing and what they were thinking while completing the task. If the participants lapsed into silence during the thinking aloud session, prompts, such as "What are you thinking now?", were used to urge the participants to verbalise their thinking. Each participant's thinking aloud session was audio-tape recorded for review after the experiment.

4.8.2 Logging

In addition to the think-aloud verbal-protocol, the participants' interactions in the spreadsheet were also logged. This logged information was employed to assist with the transcription of the think-aloud verbal-protocol. By modifying the source code of the spreadsheet, all keystrokes made by the participants were recorded in a file. This modification and recording activity did not compromise the performance of the spreadsheet and was unobtrusive during the think-aloud data gathering session. The information included: the language used, the keystroke pressed, the cell address in which the keystroke was pressed, the clock time when the key was pressed, and the final contents of the cell after

the cursor moved out of that cell. A program was written to layout the data in a more readable format. Refer to Appendix C.3 for more information on the logging data collected.

4.8.3 System Usability Scale (SUS)

The System Usability Scale was administered after the think aloud session and before the interview. The participants completed the SUS by selecting an answer to the ten questions as shown in Table 4.4.

Questions		Answers				
1.	I think that I would like to use this system frequently.	Strongly Disagree	Somewhat Disagree	Neither Agree nor Disagree	Somewhat Agree	Strongly Agree
2.	I found the system unnecessarily complex.	Strongly Disagree	Somewhat Disagree	Neither Agree nor Disagree	Somewhat Agree	Strongly Agree
3.	I thought the system was easy to use.	Strongly Disagree	Somewhat Disagree	Neither Agree nor Disagree	Somewhat Agree	Strongly Agree
4.	I think that I would need the support of a technical person to be able to use this system.	Strongly Disagree	Somewhat Disagree	Neither Agree nor Disagree	Somewhat Agree	Strongly Agree
5.	I found the various functions in this system were well integrated.	Strongly Disagree	Somewhat Disagree	Neither Agree nor Disagree	Somewhat Agree	Strongly Agree
6.	I thought there was too much inconsistency in this system.	Strongly Disagree	Somewhat Disagree	Neither Agree nor Disagree	Somewhat Agree	Strongly Agree
7.	I would imagine that most people would learn to use this system.	Strongly Disagree	Somewhat Disagree	Neither Agree nor Disagree	Somewhat Agree	Strongly Agree
8.	I found the system very cumbersome to use.	Strongly Disagree	Somewhat Disagree	Neither Agree nor Disagree	Somewhat Agree	Strongly Agree
9.	I felt very confident using the system.	Strongly Disagree	Somewhat Disagree	Neither Agree nor Disagree	Somewhat Agree	Strongly Agree
10.	I needed to learn a lot of things before I could get going with this system.	Strongly Disagree	Somewhat Disagree	Neither Agree nor Disagree	Somewhat Agree	Strongly Agree

Table 4.4: System Usability Scale

4.8.4 Interview

The last UAT employed to collect data was the interview. The objective of the interview was to gather data about the participants' opinion of the spreadsheet, as well as their suggestions on how to improve the software. The interview was semi-structured.

All the participants were asked these questions:

1. What is your opinion of the spreadsheet you have just used?
2. Can you suggest any improvements to the software?

The participants' responses in the interview were also recorded in an audio-tape.

4.9 Data Processing Procedures

This section details the steps taken to process the data collected using the UATs. The think-aloud verbal protocol and interview audio records were first transcribed. Objective/performance measures from the think aloud transcripts were then obtained. Subjective measures were obtained from SUS scores, and participants' interview comments. In addition, usability problems were also extracted from the think aloud and interview transcripts.

4.9.1 Data Transcription

All the participant's think-aloud audio records were transcribed using a data transcriber (Sony BM-80 Dictator/Transcriber with a foot pedal). The transcribed data was entered into the Speech Transcript column of the Think Aloud Transcript Log (see Table 4.5). The conventions used for the transcribed data were adapted from Ackerman et al. (1997), and are given in Appendix C.4.

Log files of participants' keystrokes were synchronised with the think aloud transcripts and used to assist the audio transcription of the think aloud data. Relevant keystroke details were entered into the Log column, column 2 in Table 4.5. To put the speech transcript into context, that is, in relation to the task being completed, the synchronised log and speech were divided into episodes. An episode was defined as a specific activity, for example, the entry of data into the spreadsheet columns. Each episode, a row in the Think Aloud Transcript Log was provided with an Episode Summary, for example "Typing data in Row 1" in Row 8 in Table 4.5.

Usability problems were then identified in the Think Aloud Transcript Log and entered into the Usability Problems column in Table 4.5. The definition of usability problems and criteria used to identify usability problems are provided in the next section.

The interview audio records were also transcribed. The transcriptions of the interview responses were typed directly into a Microsoft Word document. The think aloud and the interview audio records of every participant were examined two more times to ensure the accuracy of the transcripts. All think aloud and interview verbalisations were transcribed by the author of this thesis.

	Usability Problems	Log	Episode Summary	Speech Transcript
...	...			
8	_CR_ = _DOWN_ Used other key besides CR to enter data	Used _RT_ to enter.	Typing data in Row 1	{What are you doing now?} Okay, now masukkan data ke dalam hamparan *enter data into the spreadsheet*, I go to this side. [Typing.] Okay next...
9	Navigation using tabs. Used other key besides CR to enter data	Use tab to go to next cell. Tried _RT_ Typing in the two decimal places as well.	Typing data in Row 2	A two, rent. Oops. [Mumble] [28 secs] Two hundred.
...	...			

Table 4.5: Think Aloud Transcript Log

4.9.2 Usability Problems

To determine if the data collected from the usability evaluation will be successful, usability problems were identified from the think aloud and the interview transcripts, and then compared. A usability problem is defined as any aspect of a user interface that caused users problems "with respect to some salient usability measure (for example, learnability, performance, error rate, subjective satisfaction) and that can be attributed to a single design aspect." (Nielsen, 1993). Other criteria used to identify usability problems were:

- participant articulates a goal and cannot reach it
- participant gives up
- participant produces a result different from the task given
- participant expresses surprise or expresses negative effect
- participant says something is a problem
- participant makes design suggestion
- system crash

These criteria were obtained from Jacobsen, Hertzum and John (1998).

For example, in Episode 9 in the Log column in Table 4.5, the participant used the Tab key to move the cursor to the next cell. In this case, the use of the tab key was a usability problem in *Hamparan* as the Tab key in *Hamparan* did not work in the same way as other spreadsheets. In other spreadsheets, the Tab key

would move the cursor to the adjacent cell on the right. However, in *Hamparan* the tab key did not do anything.

After identifying the usability problems in the participants' Think Aloud Transcription Logs, the usability problems were collated. See Appendix H for the List of Usability Problems from the Think Aloud data. Usability problems were also identified from the interview data based on the criteria detailed above. The complete list of usability problems isolated from interview data is available in Appendix I.

4.9.3 Performance Measures

These behavioural measures were identified from the think aloud data and employed in the summative-process. The think aloud transcripts were scrutinised for positive and negative behaviours expressed by participants. Examples of positive behaviours included positive comments made by the participants about the spreadsheet, whereas, negative behaviours included negative comments and frustration expressed by the participants. Table 4.6 provides definitions of the positive and negative behaviour. To provide further information of participants' performance, the number of tasks completed by each participant was also identified. Refer to Appendix C.6 for procedures on determining the completion of a task and Appendix E for the number of tasks completed.

4.9.4 Subjective Measures

Subjective measures were collated from the System Usability Scale (SUS) and the interviews. This data related to the participants' opinions or views of the spreadsheet they had used. The participants' SUS responses were scored according to procedures detailed in Brooke (1996). Details of the calculations are reproduced in Appendix C.5. According to Brooke (1998), products with good usability yielded SUS scores of 60 and above. Products with bad usability scored below 50.

The interview transcripts were examined. Participants were identified as providing positive or negative responses to the question "What is your opinion of the spreadsheet you have just used?".

Positive Behaviour
Expression of positive comments: Positive comments verbalised include complimenting a particular design feature, or indicating the ease of completing a command. For instance, "... This is easy."[P12:E70]
Expression of Eureka: Participants verbalised this expression when they finally completed a task. For example, "Yay, I did it" [P3:E55], or "Yes!! I done it!" [P8:E131].
Impressed: Some participants were impressed with the spreadsheet. For example, participant P12 was impressed that he managed to enter the value into the cell. "Two hundred. Wahhhhh. [Impressed]" [P12:E21]
Negative Behaviours
Expression of negative comments: Negative comments include criticisms of the spreadsheet, such as the lack of functionality, and the lack of certain design features. "*Teruk jugak* spreadsheet *'tu!* ("Quite terrible that spreadsheet!")" [P1:E98]
Expression of frustration: During the think aloud sessions, some participants expressed their frustration quite openly. The participants' frustration can be detected through their tone of voice or their actions, such as hitting hard on the keyboard keys or knocking on the table. Expression of frustrations usually occurred when the participants were unable to complete a particular task. Expressions of frustrations also included "Tsk!" or "Hai!"
Expression of surprise: The expressions of surprise were usually verbalised when the spreadsheet did not work in the manner the participants expected the spreadsheet to. These outbursts included expressions of surprise or puzzlement such as "Oh!" or "Eh?" For instance, "Say, A six. Oh, my God!" [P1:E91] – participant thought she lost all her data.
Request for help: The participants verbalised they were looking for, or attempted to locate the help feature, especially after encountering a problem. For example, looking for function key F1, "...where's the menu? F one ah?" [P17:E29]

Table 4.6: Positive and negative behaviour descriptions

4.10 Summary

This chapter has described the procedures employed to collect data to answer the key question: Are imported UATs appropriate for usability evaluation of localised software in Malaysia, an instance of a cultural context. Section 4.1 described the adaptation of a spreadsheet utilising the global-SDLC process. An internationalised spreadsheet was designed, implemented and able to accommodate localised messages in Bahasa Melayu, Iban and Māori. Section 4.2 supplied details of an experiment in which the usability of the Bahasa Melayu version of spreadsheet was evaluated. Details on how imported UATs were assessed to determine their appropriateness in the usability evaluation were also provided. Also, the details of the experiments and procedures for collecting and processing the data collected were supplied in the remaining sections. The results of the usability evaluation and the comparison of the usability data are contained in the next chapter, Chapter 5.

Chapter 5 Results

This chapter details the results of the usability evaluation of *Hamparan*. This evaluation was conducted as part of the global-software development life-cycle (global-SDLC). The data collected using the usability assessment tools (UATs) will be used to determine the effectiveness of the imported UATs when employed in the Malaysian cultural context. The UATs are deemed effective if:

a. The data collected from them can be employed to improve the spreadsheet's usability.

b. The data collected using them can indicate the participants' perception of the spreadsheet.

These results are presented in the order in which the data were collected. Sections 5.1 and 5.2 detail the data collected using the logging and think aloud method respectively. Section 5.3 describes data collected using the System Usability Scale (SUS), and Section 5.4 reports the interview results.

5.1 Logged Data

During the think aloud session, while the participants were completing the assigned tasks, their keystrokes were recorded. This logged data was used to help in the understanding of the think aloud verbal data. For instance in "Episode 84" of participant P5's verbal data (below), participant P5 seemed to be frustrated while attempting to complete the currency format task:

> [P5:E84] S... Sunting *Edit*... [Typing] *Tak boleh!!* ("Cannot!!") [Frustrated] *Masih lagi* <D three> ("Still at <D3>") [Pause] Ummm hmmm hmmm hmmm.... [Heavy typing] [Knocking on table] [Frustrated] ...

From the logged data, it was observed that prior to this episode, participant P5 had tried 11 other commands, such as, *Baris* ("Row"), *Hamparan* ("Spreadsheet"), Goto, *Lajur* ("Column"), *Sunting* ("Edit"), and was still unable to complete the task. It was probably her many failed attempts to find the correct command that left participant P5 quite frustrated, as shown in the think aloud data.

Participants sometimes crashed the spreadsheet. By examining the logged data, the actual keystrokes prior to the crash can be identified. In the next

example, participant P8 crashed the spreadsheet at the column width prompt *Taipkan lebar lajur baru:* ("Enter the new column width:"):

[P8:E47] [Beep] Oh! [Crash #2]

[P8:Log] /-*Lajur* ("Column") -*Lebar* ("Width") -1cm-_CR_
/ [C o m m a n d] [C o m m a n d] 1 cm [Carriage return]

From the logged data, it was the "1cm" entered by the participant that crashed *Hamparan.* This "crash" could be replicated by entering "1cm" at the column-width prompt. [The acceptable input is an integer only.]

As shown in the above examples, the logged data complements the think aloud verbal data in providing a context to understand the verbal protocol. The logged data also acted as well as a source of information to specific problems the participants experienced while using *Hamparan.*

5.2 Think Aloud Data

The think aloud data comprised the participants' verbalisations recorded when they were using the spreadsheet. Think aloud was employed to collect data which could be applied to improve the usability of the spreadsheet and, in this experiment, was used to collect data about the participants' perception of *Hamparan.* In relation to the formative-process, usability problems were identified from the think aloud data. The think-aloud usability problems are data which could be used to improve *Hamparan's* usability. These usability problems identified are detailed in Section 5.2.1. In relation to the summative-process, two types of user behaviours were identified in the think aloud data to indicate the usability of the software. These behaviours were negative and positive behaviours. Examples of negative and positive behaviours are reported in Section 5.2.2. Actual counts of these behaviours are also detailed in Section 5.2.2.

5.2.1 Usability Problems from Think Aloud Transcripts

This information is identified from the think aloud transcripts as usability problems. A usability problem is defined as any aspect of a user interface that caused the participants problems.

Sixty-three usability problems were identified from an examination of all 17 transcripts of think-aloud verbal data. An example is when the participant did

not know the syntax of the formula to sum a range of cells. This problem is illustrated by a participant's comment while attempting the Formula Task. Participant P13 tried to complete the task by entering "@SUM(B4.B6)". She noted that the attempt was unsuccessful:

[P1:E13] Okay, ahh... so we have to type this one, how do you say this one? [Pointing to the @ on the keyboard] {Alias} Alias... S U M bracket... and then type B... four and then... full-stop and then... B... six. And then close the bracket... Cannot...

[P1:Log] @SUM(B4.B6) [formula entered]

(The "@" symbol is known as "alias" in Malaysia (Merriam-Webster, 1997).) Every think aloud usability problem identified is documented in the form shown in Table 5.1. Each usability problem is given a number, in this case, No. 35 and described. The criterion used to identify this usability problem is supplied as well as evidence of the problem. The complete list of problems identified is available in Appendix H.

35. Formula syntax was not intuitive

Description: The word "Intuitive" here means intuitive to participants who have used spreadsheets before. As the participants have had no experience using the FIRST spreadsheet, the formula syntax in spreadsheet were not known to them. The correct formula to sum cells B2, B3 and B4 are B2+B3+B4 and B2:B4.

Criteria: Participants commented they are unable to complete the formula or participants tried several types of formulas.

Evidence:

[P1:E14] ... Alias... S U M bracket... and then type B... four and then... full-stop and then... B... six. And then close the bracket... Cannot...

[P1:L] @SUM(B4.B6)

[P3:L] =sum(b2..b4)

[P15:L] =TOTAL(B2..B4)

Table 5.1: Usability Problem No. 35: Formula syntax was not intuitive.

The formula syntax problem was encountered by the most number of participants (13 participants), see Table 5.2. Given the high number of participants experiencing this difficulty, this usability problem may be given a

high priority to be corrected in the next version of the software. In addition to the formula syntax problem, there were eight other usability problems which were encountered by more than half of the 17 participants.

Usab. Prob. No.	Usability Problem Name	No. of participants faced the problem	Proportion of 17 participants
35	Formula syntax not intuitive	13	76%
17	Unit of the cell width was not known	12	71%
30	Do not know meaning of Auto	12	71%
34	Need to press Enter or other arrow key to go to next cell	12	71%
48	Mistook English as default instead of Bahasa Melayu	10	59%
2	Used arrow keys to select the command	9	53%
10	Difficult to interpret prompt "Enter the first cell you wish to format:"	9	53%
44	Treated DOWN and UP arrow key as CR key	9	53%
46	Had to translate Bahasa Melayu terms to English	9	53%
16	Do not know meaning of Lajur ("Column")	8	47%
37	Navigation using tabs.	8	47%

Table 5.2: Usability problems faced by participants in the think aloud session

5.2.2 Participants' Perception of *Hamparan*

Many negative behaviours were identified in the participants' think aloud data. Negative behaviours comprised negative comments, expression of frustration, expression of surprise, and expression of seeking help. These negative behaviours are illustrated in the context of the participants' attempts in completing the assigned tasks. These behaviours indicate the participants' perception of *Hamparan*.

Negative Behaviours

Some negative comments were made by participants at the beginning of their interactions with *Hamparan*. For example, participant P3 remarked negatively about *Hamparan*'s interface:

> [P3:E6] Oh mmm okay... so this is all that you can do. I don't like your interface, Alvin. [laughs] Too primitive! I mean not to say the screen or what, I mean it reminds me of aaa Lotus 123.

Participant P3 also made a negative comment about the spreadsheet, that is, about the lack of mouse use:

[P3:E7] Oh my God. Alvin, Alvin, Alvin [tsk, tsk, tsk] You need to use mouse.

Participant P8's initial impression of *Hamparan* was that he preferred a window interface (graphical user interface) instead of the DOS interface (command user interface) of *Hamparan*:

[P8:E9] I feel, I feel, I... it's better if I use a window because this DOS [spreadsheet] is not really effective... I think so... [Laughs] Because we have to... face to a black mirror. [Laughs]

The "black mirror" probably referred to *Hamparan*'s cells, which were black.

- Data Entry

In the Data Entry tasks, the participants were asked to enter a table of data. This table can be seen in Section 4.5. Participant P13 complained on the slowness of entering the data into the spreadsheet:

[P13:E4] Uh hum Slow lah this one [Entering data] Okay, March. [Typing "March"]

- Creating a Formula Task

In this task the participants were required to create a formula that summed the expenses for the months of January to March. Negative comments also included remarks about the difficulty in completing the task. In this instance, participant P11 was at a loss as to how she should create the formula:

[P11:E22] *Macam mana nak guna?* ("How to use/create it [the formula]?") [Laughs] I *tak biasa.* ("I am not familiar [with the formula/spreadsheet]")[whisper]

Participant P16 commented on the lack of help, another negative comment made about *Hamparan.*

[P16:E20] Right... now, how do... should I total up this? See? Yuu! *Saya tak pandai pakai* this type of thing. ("I do not know how to use this [spreadsheet]") [Laughs] Very <> no helps

- Save Task

In this task, the participants were requested to save the spreadsheet document they had created. Participant P15 made a critical comment about the lack of feedback as to whether the file was saved:

> [P15:E61] So I can... so I just key in my file name and there's no indication telling me that the file has been saved or not.

- Column Width

In this task, the participants were required to widen the column so that the full word "Electricity" was displayed in the spreadsheet. "Electricity" is 11 characters long whereas the default column width of the spreadsheet is only 10 characters. Participants remarked about the difficulty in completing this task:

> [P8:E42] ... I think is... very difficult, to use this one *Lebar* ("Width [command]")

> [P13:E52] *Aiyaaa banyak susah ini benda...* ("This thing very difficult") [Laughs]

Participant P11 expressed her frustration after crashing the spreadsheet twice in completing the column-width task.

> [P15:E56] *Dua-dua, dah dua kali cuba tiga kali dah dia keluar balik ke situ.* ("Twice, three times, I try, and I exit [crash] as well.") {*Cuba, cuba* ("Try try")} *Cuba lagi...* ("Try again") [sucking in breath] *Kalau tak dapat?* ("If cannot?") [Frustrated tone]

- Currency Format

In this task, participants were asked to change the format of the total figures to RM format. [RM (Ringgit Malaysia) is the name of the Malaysian currency.] In this task, many participants had difficulty in answering the prompt *Masukkan sel pertama untuk diformat* ("Enter the first cell to format"). The correct input to the prompt is a cell address, which specifies the first cell of a range of cells to be formatted.

Participant P9 typed "Rm", "rm$", and "RM$" at the prompt. The participant then tried "RM55", and pressed Enter which caused the error beep ([Beep]). Her frustration was quite evident:

[P9:E78] R M... [Typing.] [Beep!] Hai! [Frustrated]

Similarly, participant P5's frustration was noticeable after she had tried eleven other commands such as *Baris* ("Row"), *Hamparan* ("Spreadsheet"), Goto, *Lajur* ("Column"), *Sunting* ("Edit") and was still unable to complete the task:

[P5:E84] S... Sunting *Edit*... [Typing] *Tak boleh!!* ("Cannot!!") [Frustrated] *Masih lagi* <D three> ("Still at <D3>") [Pause] Ummm hmmm hmmm hmmm.... [Heavy typing] [Knocking on table] [Frustrated] ...

While attempting this task, participant P4 also complained about the difficulty in completing the task:

[P4:E10] It's quite /difficult.\

Participant P13 remarked negatively about the lack of clarity of the prompt:

[P13:E32] How is it? *Tak clear lah* this one, [irritated] *sel pertama untuk diformat, masukkan sel pertama untuk*... ("*This one* [prompt] is not clear, enter the first cell to format.")

- Quitting the Spreadsheet

Just before quitting the spreadsheet, participants also commented critically about the spreadsheet. Such comments reflect the participant's negative opinion of the spreadsheet, for example participant P1:

[P1:98] *Teruk jugak spreadsheet tu!* ("Quite terrible that spreadsheet!")

Positive Behaviours

While there were many negative behaviours that were isolated from the think aloud data, there were fewer positive behaviours. Examples of the positive behaviours include:

- Positive Comment

Participant P12 commented at the ease of entering the data into the spreadsheet:

[P12:E70] Okay 'kay, *kita edit ya kita sunting* ("yes, we edit yes we edit") *S, Sunting.* ("S, Edit") This is easy. Electri-c- Electri-T-R-I-C- ... *betul* ("Correct?") Elec... Elec... Okay..

- Expressions of Eureka

Expressions of Eureka were mainly verbalised when the participants successfully completed a task or part of a task. For example, participant P3 had successfully completed the Currency Format task:

[P3:E55] *Adakah anda semua... bercomma?* ("Do [you want] commas in numbers?") *Boleh...* ("Can...") Yay. I did it. [Eureka]

- Impressed expressions

Participants at times were impressed with the spreadsheet. Participant P12 was "impressed" that he managed to enter the value into the cell.

[P12:E21] Two hundred. Wahhhhh. [Laughs]

While the above expression may be deemed as sarcastic, the author interprets the above expression of being impressed. Expressions of "wah", made by Malaysians, can be interpreted as surprise, impressed and disappointment (Gaudart, 1995; p.33).

	Positive Com-ments*	Eureka*	Impres-sed*	**Total Count Positive Behaviour**	Negative Com-ments	Frus-tration	Unexpec-ted	Help	**Total Count Negative Behaviour**	No. of tasks completed	Crash
P1	0	1	0	**1**	4	21	8	0	**33**	3	4
P2	0	0	0	**0**	4	6	5	1	**16**	4	1
P3	0	5	2	**7**	8	1	4	0	**13**	5	0
P4	0	0	0	**0**	2	2	2	0	**6**	5	0
P5	1	0	0	**1**	4	27	6	0	**37**	4	0
P6	0	5	0	**5**	8	7	2	0	**17**	5	0
P7	0	0	0	**0**	0	8	3	0	**11**	1	4
P8	0	3	0	**3**	14	12	5	0	**31**	3	3
P9	0	0	0	**0**	1	12	6	0	**19**	2	1
P10	0	6	0	**6**	0	3	18	0	**21**	5	0
P11	0	2	0	**2**	2	12	7	0	**21**	3	2
P12	6	4	1	**11**	4	4	7	0	**15**	6	0
P13	1	3	0	**4**	11	15	2	0	**28**	4	0
P14	1	0	0	**1**	1	8	6	0	**15**	3	0
P15	1	0	0	**1**	11	1	2	1	**15**	5	1
P16	3	3	2	**8**	3	3	1	3	**10**	5	1
P17	0	3	0	**3**	2	7	1	1	**11**	4	0
Total	13	35	5	**53**	79	149	85	6	**319**		17

Table 5.3: Counts of positive & negative behaviours, crashes & tasks completed
*Positive behaviours are marked with an asterisk *. The remaining behaviours are negative behaviours.*

Based on an inspection of all 17 think aloud transcripts, there were more negative than positive behaviours. This observation is supported by the counts of the behaviours as shown in Table 5.3. From the table, only 13 positive comments where identified while 79 negative comments were found. In addition, the number of expressions of frustration was 149, nearly three times the total number of positive behaviour, that is, 53.

The data in Table 5.3 shows that the majority of the participants had difficulty using *Hamparan*. The high number of negative behaviours compared to the low numbers of positive behaviours supports this interpretation. However, one particular participant found the spreadsheet relatively easy, participant P12. He was the only participant who completed all tasks, and the Bahasa Melayu interface was quite intuitive to him as shown by the high number of positive behaviours in Table 5.3.

5.3 System Usability Scale

The SUS was employed to garner the participants' opinion of whether *Hamparan* possesses good or poor usability. In Table 5.4, the SUS responses have been standardised, compensated for reverse polarity, such that all responses of 1 are most unfavourable whereas 5 are most favourable. The values 1 (most favourable) to 5 (most unfavourable) are used as nominal values and not as ordinal numbers associated to magnitude of values. A score of 3 means the participants neither agreed nor disagreed with the statements. The SUS scores were calculated according to instructions provided in Brooke (1996). Participants' SUS scores of 60 and above, indicate that these participants rated the spreadsheet as possessing good usability (Brooke, 1998). Participants' SUS scores which are below 50, indicate that the spreadsheet has poor usability (Brooke, 1998).

Out of a possible score of 100, the participants' SUS scores ranged from 22.5 to 95.0. From Table 5.4, the participants' average System Usability Scale score for *Hamparan* was 50.0 and the median, 47.5, which indicate *Hamparan* as possessing bad usability. Nine (53%) of the 17 participants rated the spreadsheet as having poor usability. Only three (18%) of the 17 participants scored the spreadsheet as possessing good usability. Given the high number of participants

who rated the spreadsheet as having poor usability, as well as the low average and median scores, the participants had indicated that *Hamparan* has poor usability. The complete scores of each participant are available in Appendix F.

Participant	Status	Familiarity	Gender	SUS Scores	Usability
P12	High	Yes	Male	95.0	Good
P3	High	Yes	Male	72.5	Good
P16	High	Yes	Female	67.5	Good
P5	High	No	Female	55.0	Neutral
P2	Low	Yes	Male	52.5	Neutral
P4	Low	No	Female	52.5	Neutral
P7	Low	Yes	Female	52.5	Neutral
P9	Low	No	Female	52.5	Neutral
P6	High	No	Male	47.5	Bad
P10	High	Yes	Female	47.5	Bad
P14	Low	No	Male	47.5	Bad
P1	Low	Yes	Female	42.5	Bad
P13	High	No	Female	40.0	Bad
P15	High	Yes	Male	40.0	Bad
P11	High	Yes	Female	35.0	Bad
P17	High	No	Male	27.5	Bad
P8	Low	No	Male	22.5	Bad
			Mean	50.0	
		Grand Mean	50.00	Min.	22.5
		Median	47.50	Max.	95.0
		Stdev	17.05	Lower Quartile	40.00
		SEMean	4.13	Upper Quartile	53.75

Table 5.4: System Usability Scale scores (sorted in descending order)

5.4 Interview

The interview was employed to collect data to meet the two aims. The first aim was to collect information that could be used to improve the spreadsheet. This information is provided in Section 5.4.1. The interview was also applied to garner the participants' opinions of *Hamparan* for the second aim. These opinions are described in Section 5.4.2.

5.4.1 Usability Problems from Interview

In the formative-process, the UATs should be able to collect data which in turn can be applied to improve the spreadsheet. To collect this data, the

participants were asked, "What suggestions can you give to improve the interface/spreadsheet?". The participants' suggestions were isolated from the interview transcripts. These suggestions are presented as usability problems since these suggestions are actually problem areas that the participants thought should be corrected or improved.

For example, five participants, P1, P3, P8, P9, P17, commented on the clarity of the commands. The suggestions are detailed in Table 5.5 and are grouped under Interview Usability Problem No. 7.

7.	**Commands were not clear**
The participants commented that the commands were not clear and that the instructions/commands must be clear.	
[P1:Q2]	Simplify the words [commands]
[P3:Q2]	... I think improvements to the software, some of the words [command] were not really clear what they mean.
[P8:Q2]	... It need to... umm, need to make an improvement hmm... what's hmm... aaa... very tsk [frustrated]...ahhh, ... *arahan-arahan yang lebih mudah lah.* ("commands that are much simpler")
[P9:Q2]	... *Macam* ("Like")... ahhh... *word yang senang sikit* ("words which are easier?") {*Maknanya word 'tu... makna word tu arahan kan?* ("By word you mean commands?")} *Arahan, ya ya arahan* ("Commands yes, yes, commands.") ...
[P17:Q2]	In terms of Bahasa Melayu version just now... I think that the lettering or the command is quite ahh... confusing. ...

Table 5.5: Interview Usability Problem No. 7: Commands were not clear

Another suggestion given by the participants was to provide help:

[P6:Q2] ... or there must be a help function somewhere... so that the user can can understand. Just, like just now, I am trying get to get through the formula... I don't know where to go [Laughs]

By providing users with the help function, the participants would then be able to use the help feature to assist them in using *Hamparan*. This suggestion was identified as Interview Usability Problem No. 2.

There were 20 usability problems that were identified from the interview transcripts. The problems most frequently encountered were: "help was not available" and "commands were not clear", see Table 5.6. Five participants noted

those two usability problems. The full list of the interview usability problems can be obtained in Appendix I.

Usability Prob. No.	Usability problems	Number of participants who noted problem	Percentage of 17 participants
2	Help was not available	5	29%
7	Commands were not clear	5	29%
4	Looked for mouse	3	18%
6	Numerous functions were unavailable	3	18%
9	Confusion between yes/no and ya/tidak	3	18%
12	Lack of GUI	3	18%
1	Slow Performance	2	12%
8	Bahasa Melayu terms problematic	2	12%
14	Did not know how *Tunjukkan Rumus* ("Formula Display") command worked	2	12%
11	Formula syntax unclear	1	6%

Table 5.6: Usability problems identified in interviews

5.4.2 Participants' Opinion of *Hamparan*

To elicit opinions about the spreadsheet, the participants were asked, "What do you think of the spreadsheet you have just used?". There were seven participants whose responses were deemed as positive. The complete list of the interview responses is available in Appendix G.

Positive Responses

The comments listed below provide instances of participants' positive opinions. From these comments, some participants appeared to be very positive about the software. Participant P12, was so positive that he believed that the spreadsheet should be commercialised.

> [P12:Q1] *Bagus! Senang digunakan. Boleh komersilkan, saya rasa.* ("Good! Easy to use. Can be commercialised, I believe.")

Participant P9 was also positive about *Hamparan*. She considered *Hamparan* to be good enough to replace Excel as she deemed *Hamparan* to be easier than Excel.

> [P9:Q1] Ummm... *hamparan itu bagus ya untuk umm* for umm... *mengganti... menggantikan Excellah* ("The spreadsheet is good... replace... replacing Excel...") {*Mengapa?* ("Why?")} *Sebab dia* ah... *bagaimana nak cakap* [Snigger] *Lebih senang dari Excel lah.* ("Because it... ah... how to say this... easier than Excel.")

Participant P4 echoed this positive sentiment. P4 thought that *Hamparan* was similar to Lotus 123 and she felt that *Hamparan* was better than Lotus 123. Participant P4 added that *Hamparan* was needed to replace Excel as she and her colleagues experienced problems with their calculations in Excel.

[P4:Q1] I think it's okay, it's more, more like Lotus. I think we better use this in [department] there. ... I got the feeling it's much more better than Lotus, it's quite difficult at the first time. ... It's aaa more like Lotus, so I think we need this. Because sometimes we use Excel, we got problems with our calculations...

Although participant P16 believed that is was "quite difficult" for beginners to use the spreadsheet, she remarked that it would be "very easy" if the user was familiar with spreadsheets. Participant P16 was deemed to have found the spreadsheet to be easy. She was a systems analyst and thus had more experience using computers.

[P16:Q1] For a beginner, it's quite difficult for them to learn this... to use this system {Okay...} Unless you are familiar with ahhh... other spreadsheets... then it should be very easy.

Negative Responses

The following comments below provide instances of participants' negative opinions of the spreadsheet. Ten participants were ascertained to have negative perceptions of the spreadsheet. Participants P3, P6, P15, P13 and P17 were very outspoken about their views of *Hamparan*. As shown below, participant P3 and P15 were critical of the spreadsheet:

[P3:Q1] Umm, old fashioned, umm, reminded me of Lotus 123.

[P15:Q1] I think the system still at a very primitive level... {ummm...} by providing some... umm... what do you call basic function {okay} and there are still a lot of room to improve the system...

Participant P13 was also forthright with her comments. When asked her opinion about the spreadsheet, participant P13 replied immediately that she did not like the spreadsheet:

[P13:Q1] [Without hesitation] I don't like it ...

Participant P17 was candid in his comments; he thought that the spreadsheet was really difficult to learn. However, he mitigated his comments by saying that the spreadsheet was not that difficult provided he had "pre-testing".

[P17:Q1] \Spreadsheet/ I think it's really difficult to learn. Ahh... no, actually its not... not that difficult... but I need to have this pre-testing ahh... {Ahh... okay} So that I... I can use it... because it's happened that I've used this thing quite some time before. I have to practise it... then I can... can make use of it. {Okay}

Participant P10 remarked negatively about *Hamparan*. Participant P10 disclosed that she wanted to "enjoy" using the spreadsheet. However, she was shocked at the lack of commands and got *pening* ("a headache"); the spreadsheet was so difficult that it gave her a headache. She also commented that the spreadsheet was not user friendly.

[P10:Q1] What was my opinion? [Laughs] I wanted to enjoy it but I was like ... ahhh shocked... because there's, like, nothing there.. nothing... kind of *arahan* ("commands") until I really got *pening* ("a headache"). I guess usually... in any kind of arr... < >, have to be friendly, when I opened it, it was like... I was taken aback... It wasn't that friendly I have to be frank with you... [Laughs]

Two participants, P5 and P7, had the opinion that it was much easier to use Excel than *Hamparan*.

[P5:Q1] I think *mudah guna Excel* ("easier to use Excel.") [Laughs]

[P7:Q1] Umm... I think *Excel lebih senang lagi* ("Excel is [much] easier...")

Participant P11 responded that using *Hamparan* was difficult as she was not familiar with it.

[P11:Q1] [Pause] *Menyukarkanlah...* ("Difficult...") *Menyukarkanlah... sebab saya tak biasa...* ("[Pause] Difficult... because I am not familiar... [to the spreadsheet]")

Participants also gave reasons why they disliked the software. Participant P5 said the spreadsheet was "slow" as she had to type in formula as opposed to just clicking the AutoSum button in Excel:

[P5:Q1] ... And then this one, sometime you have to create your formula <> kan? Right? Takes a long time lah. Since Excel you just click with the formula, sum...

Participant P1 also said the spreadsheet was slow, as she had to type '/' to access the commands.

[P1:Q1] [interview notes] Got to have slash before instruction appear on the screen. Not fast, have to go slash..

Some participants compared *Hamparan* with spreadsheets they had used before. There was a lack of functionality in *Hamparan* when compared to spreadsheets such as Excel. Examples of functions that the participants looked for included: access to help, use of the mouse, and a summation button which enters a formula to sum the values in a column or row.

5.5 Summary

This chapter has described the data obtained using the three UATs. The data collected from the logging augmented-think aloud, SUS and interview were employed to provide an indication of the participants' opinion of the spreadsheet they had used. The data collected using think aloud and interview were also intended to assist with improving *Hamparan's* usability. In the next chapter, these results are analysed to determine whether the imported UATs are effective when employed in Malaysia.

Chapter 6 Analysis

The last chapter described the data collected by the imported UATs in the usability evaluation of the localised spreadsheet, *Hamparan*. This chapter analyses the data collected. These analyses are intended to help decide whether the imported UATs were effective if the data collected can be:

a. Applied to improve the usability of *Hamparan*

b. Employed to ascertain the usability of *Hamparan*

The analysis with regard to improving usability of *Hamparan* is presented in Section 6.1, whereas the analysis with regard to determining usability of *Hamparan* is given in Section 6.2.

6.1 Usability Problems: Think Aloud and Interview

The first aim of the usability evaluation was to ascertain whether the imported UATs could collect usability data, which could then be employed to improve *Hamparan*. The usability data were identified from the think aloud and interview transcripts as usability problems. Analyses and aggregation of the think aloud and interview usability problems are reported in the next section, Section 6.1.1. These usability problems were classified into categories. These categories were used to provide further perspective on the type of problems experienced by the participants and can be used to identify possible solutions to the problems. This classification is presented in Section 6.1.2.

6.1.1 Specific usability problems and comparison

The analysis of the usability problems, in particular the comparison between the usability problems from the think aloud and interview data, was conducted to ensure that the usability problems were confirmed in the data. Consistency between the think aloud and interview usability problems would cross-validate data collected using the two UATs.

Sixty-three usability problems were identified from the think aloud transcripts. Twenty usability problems were identified from the interview responses. There was substantial agreement between the usability problems found in the think aloud data and those identified from the interview data. This

agreement of usability problems identified from the two sources is shown in Table 6.1.

An example is provided to illustrate this concurrence, that is, between Think Aloud Usability Problem No. 8 and the Interview Usability Problem No. 17. In the think aloud session, participant P15 noted a message *Dalam proses memuatkan fail* ("Loading file...") was displayed too briefly:

> [P15:E62] There was a message... but I didn't see it clearly, the system... showing so fast and disappear like this...

Usability Problems from Think aloud Data		Usability Problems from Interview Data	
UP#	Usability Problem	UP#	Usability Problem
1	"/" Slash key was not clear.	20	Press / instruction was not clear
8	Message "Loading file..." displayed too briefly	17	Message appeared too briefly
17	Unit of column width was not known	13	No information about the column width unit
27	*Tunjukkan Rumus* ("Formula Display") mode was not obvious	14	Did not know how *Tunjukkan Rumus* ("Formula Display") command worked
35	Formula syntax was not intuitive	11	Formula syntax unclear
38	Expected carriage return to move to next cell	15	Enter key did not work as expected
47	Not sure to answer yes/no or *ya/tidak* .	9	Confusion between yes/no and ya/tidak
49	Looking for Help	2	Help was not available
51	Looked for mouse.	4	Looked for mouse
54	Looked for AutoSum	5	Looked for summation buttons (AutoSum)
55	Looked for a way to select/highlight cells	16	Could not select cell range
57	Looked for filename on spreadsheet	18	Expected file name on window
59	Looked for Windows	12	Lack of GUI

Table 6.1: Matching think aloud and interview usability problems
UP#: usability problem number in List of Think Aloud Usability Problems and List of Interview Usability Problems.

Participant P15 later identified this problem in the interview, and pointed out that the message appeared too briefly. Participant P15 also recommended ways to fix the problem.

> [P15:Q2] ... There are some improvements to be done because the message appear too fast... [too briefly] Urr... the user cannot get the message... I think this is even worse if you run the system on a faster machine [Laughs] So... the system should give some delay or just a pop up message and wait for the stu... wait for the user response urr... by pressing a key to confirm the message... so that the user know what happening to them.

Of the 20 interview usability problems, 13 could be associated with a specific think aloud usability problem. The usability problems identified from the interview data were more "high level/global" (see Table 6.2) and they encompassed a number of specific think aloud usability problems. For example,

the Interview Usability Problem No. 6, "Numerous functions were unavailable" would include think aloud usability problems such as "Looked for AutoSum", "Looked for Undo", "Looked for Copy", that is, functions that participants looked for during the think aloud session.

There were only two interview usability problems that were not associated with any think aloud usability problem identified. These interview usability problem, were Problem No. 3, "Could not find menus" and No. 10, "Unsure how to answer prompt, Y or enter". Interview Problem No. 10 would probably be rectified by providing extra information about what the valid inputs are. As for Interview Usability Problem No. 3, participant P5 suggested locating the menus at the top of the screen like Excel. This suggestion would be an improvement to accessing the menus using the '/' key.

From the above analysis, the think aloud and interview usability problems cross-validate each other. This validation provides greater confidence in the reliability of the usability problems.

Usability Problems from Interview Data	
UP#	Usability Problem
19	Menu navigation was not effective
7	Commands were not clear
1	Slow Performance
8	Bahasa Melayu terms problematic
6	Numerous functions were unavailable
17	Message appeared too briefly

Table 6.2: "High-level/global" interview usability problems

6.1.2 Categories of Usability Problems

The think aloud and interview usability problems were grouped into the categories identified as usability heuristics from Nielsen (1993). These heuristics are basic characteristics of a good usable interface (Nielsen, 1993). Problems identified under these categories can then be improved using the principles associated with that category. For example, one of the most basic usability principles is consistency; the same information should be located at the same place on the screen, and commands should invoke the same actions (Nielsen, 1993). With regards to the think aloud data, 30% of the 63 think aloud usability problems, and 50% of the 20 interview usability problems fell into the category "Inconsistencies with other Spreadsheets", refer Table 6.3. These usability

problems pertain to the inconsistency of functions of *Hamparan* when compared with spreadsheets that the participants were familiar with. Spreadsheets that the participants were familiar with included Lotus 123 or Microsoft Excel. The participants expected to find functions like Undo (Think Aloud Usability Problem No. 53) and AutoSum (Think Aloud Usability Problem No. 54), which are available in more well-known spreadsheets and would have assisted them in completing common spreadsheet tasks. However, these functions were not available in *Hamparan*. The problems in this category could be improved by ensuring *Hamparan* provides the same or similar functions as other common spreadsheets. For example, in the think aloud usability problems, participants were looking for the "undo" command. Thus, to be consistent with other spreadsheets, the "undo" command should be included in the new version of the spreadsheet. *Hamparan* would be more acceptable to participants since the participants would be provided with frequently used functions, which are available in other spreadsheets.

Categories		Number of Usability Problems			
		Think Aloud	Proportion of 17 participants	Interview	Proportion of 17 participants
1.	Inconsistencies with other spreadsheets	19	30%	10	50%
2.	Unclear commands	13	21%	1	5%
3.	Spreadsheet Problems	9	14%	1	5%
4.	Unclear dialogue messages/prompts	7	11%	2	10%
5.	Poor feedback	6	10%	2	10%
6.	Language problems	4	6%	2	10%
7.	Mode problems	2	3%	1	5%
8.	No clear exits	1	2%	0	0%
9.	No help	1	2%	1	5%
10.	Shortcut problems	1	2%	0	0%
	Total	63	100%	20	100%

Table 6.3: Categories and number of TA and interview usability problems

The second highest number of think aloud usability problems was grouped under the category "Unclear commands". This category of usability problems was concerned with commands that the participants did not understand or which they did not know the meaning of the computing terminology in Bahasa Melayu. Some commands that the participants failed to understand include *Baris* ("Row") (Usability Problem No. 19), and *Lajur* ("Column") (Usability Problem

No. 16). In this example, the heuristic associated with this category is "speak user's language", that is, software engineers should use language or terms familiar to the participants. Since the participants were new to Bahasa Melayu computing terminology, and that the participants have had experience using other spreadsheets, a function could be added so that every time a Bahasa Melayu command were selected, an English translation of the command would be displayed on the screen. This feature could be useful for spreadsheet users who are migrating from using spreadsheet with an English interface to one with a Bahasa Melayu interface. It could also be useful for users who have only used spreadsheets with a Bahasa Melayu interface. These users may find the English commands useful should they need to use a spreadsheet application that only allows interactions in English.

As shown in the above examples, the categories provide some guidelines on how to improve or fix the problems identified from the think aloud or interview data. By correcting the usability problems, the usability and, in turn the acceptability, of the spreadsheet could be improved.

Another observation made of the identified usability problems is that the think aloud problems were quite detailed when compared with the interview problems. While these two sources of usability problems agreed substantially, the interview problems, as shown in Table 6.2, were not as specific as the think aloud problems. The think aloud problems (see Appendix H) were evidenced by the participants' think aloud data and/or logged keystroke data. Thus, information of the think aloud usability problems was quite detailed. Some problems, such as spreadsheet crashes, could be replicated based on the logged keystrokes. A software engineer would be able to use this information to fix the usability problem.

On the other hand, the interview responses were "high-level/global" as shown in Table 6.2. It should be noted that there were some interview usability problems which were as specific as the think aloud problems shown in Table 6.1. Although the think aloud data is more detailed, the interview information does complement the think aloud information. As reported in Section 5.2.1, the think aloud usability problem experienced by most participants was the problem, "formula syntax not intuitive". Only one participant in the interview identified

this usability problem. However, in the interviews, "lack of help" was noted by most number of participants. It is possible that the addition of help documentation would assist users in obtaining the correct formula syntax problem, as some participants such as participant P16, searched for help to complete the formula task:

> [P16:E24] Ummm.... okay, as the best way... to look for the formula... [Pause] Then find... then have to get help from the F one key, the help is not provided.

Lists of all the usability problems and the categories in to which they were divided are given in Appendix J.1 (think aloud categories), and Appendix J.2 (interview categories).

6.1.3 Cross-validation of Usability Problems

Substantial agreement between the usability problems identified from the think aloud and interview transcripts cross-validated the usability problems from these two sources (see Table 6.1). Details of the usability problems, especially the think aloud usability problems, appear to be sufficiently detailed for a software engineer to rectify them. Categories into which the problems were grouped could also indicate how to improve the usability of *Hamparan*. The formative evaluation process of the imported UATs successfully collected data to improve the spreadsheet – the first aim was achieved.

6.2 Participants' Perception of *Hamparan*: Analysis

This section analyses results with respect to the second aim of the usability evaluation. This aim pertains to ascertaining whether imported UATs employed in the summative process can collect data. The data collected can then be employed to gauge the participants' perception of the spreadsheet. This aim is achieved if the data collected using the UATs can be employed to determine the usability of the tool.

In the think aloud data, the number of negative behaviours observed far exceeded the number of positive behaviours isolated from the think aloud-verbal data. As shown in Figure 6.1, 62% of behaviours identified were frustrations and negative comments expressed about the spreadsheet. These behaviours suggest

that the participants had difficulties using *Hamparan* and very likely had a negative perception of *Hamparan*.

This think aloud result is supported by the data from the participants' SUS ratings. More than half of the 17 participants rated *Hamparan* as possessing poor usability while only three participants rated the spreadsheet as having good usability. About a quarter of the participants rated the spreadsheet as neutral. A breakdown of the SUS scores is depicted in Figure 6.2.

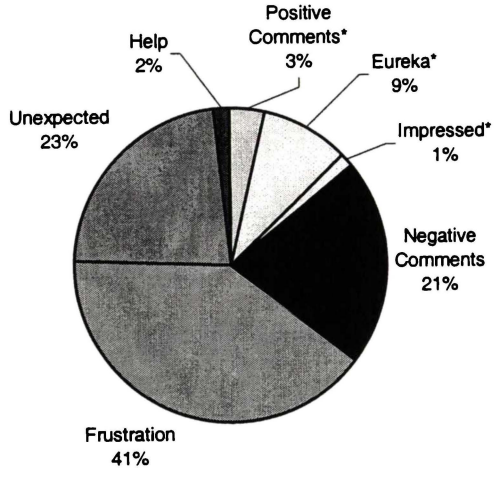

Figure 6.1: Distribution of the behaviours isolated from the think aloud data
*Positive behaviours are marked with an asterisk *. The remaining behaviours are negative behaviours.*

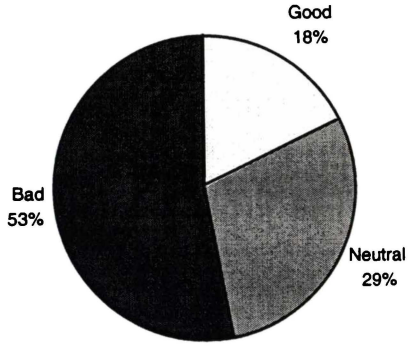

Figure 6.2: Proportion of participants' SUS ratings of *Hamparan*'s usability

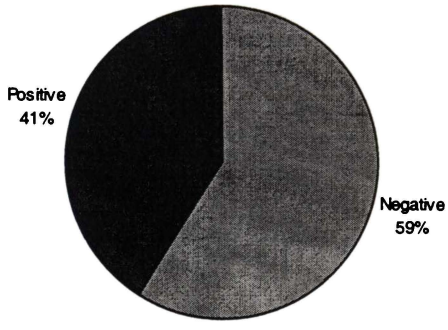

Figure 6.3: Proportion of participants' opinion of *Hamparan*

The participants' perception of *Hamparan's* poor usability was also supported by the participants' interview responses. Ten (59%) of the 17 participants' opinions of *Hamparan* were deemed to be negative opinions, see Figure 6.3. Their negative comments include: "old fashioned" [P3], "very outdated... very basic" [P6] and "very primitive" [P15]. On the whole, the interview data indicated that participants had a negative opinion of *Hamparan.*

Based on an examination of the data collected by employing the three UATs, participants, in general, viewed *Hamparan* as possessing poor usability.

6.2.1 Aggregation of the Data

From the overall results in the previous section, it would appear the results of the three UATs were consistent with one another. However, on closer inspection of the aggregated data in Table 6.4, there appears to be inconsistencies in the UAT data at the participant level. The data from the three UATs in Table 6.4 are sorted from most positive to the most negative of the aggregate think aloud (TA) behaviour score of each participant.

The TA scores were used because the think aloud data, augmented by logged data, was considered as a more objective measure than SUS (a questionnaire) data and interview responses. Each participant's aggregate TA behaviour score is calculated by subtracting that participant's number of negative behaviours from the participant's number of positive behaviours. This order suggests that think aloud data of participants in the upper rows of Table 6.4 comprised more positive behaviours or fewer negative behaviours as compared to

TA Beh.	SUS Usab.	Inter-view	Sta-tus	Fam iliar	Gen-der	Tasks compl	Parti-cipant	Interview Responses
-2	Good	Positive	High	Yes	Female	5	P16	... Unless you are familiar with ahhh... other spreadsheets... then it should be very easy.
-4	Good	Positive	High	Yes	Male	6	P12	("Good! Easy to use. Can be commercialised, I believe.")
-6	Neutral	Positive	Low	No	Female	5	P4	I think it's okay, it's more, more like Lotus. I think we better use this in [department] there. ... I got the feeling it's much more better than Lotus, it's quite difficult at the first time. ...
-6	Good	Negative	High	Yes	Male	5	P3	Umm, **old fashioned**, umm, reminded me of Lotus 123.
-8	Bad	Negative	High	No	Male	4	P17	\Spreadsheet/ I think it's really difficult to learn. Ahhhh... no, actually its not... not that difficult... ...
-11	Neutral	Negative	Low	Yes	Female	1	P7	[Deleted] ("It's good for daily work... and for saving data...") ... Umm... I think Excel ("Excel is easier")...
-12	Bad	Negative	High	No	Male	5	P6	Of course, if you compare with the... current spreadsheets we have in the market, of course that one is **very outdated**. ... Now, spreadsheet is is very advanced, ... So this one is **very basic** lah, in a sense. ...
-14	Bad	Positive	Low	No	Male	3	P14	("I feel that the spreadsheet... I feel it... if we learn... it's easier to use... and it... easier for use to complete all sorts of [tasks] that we want.")
-14	Bad	Negative	High	Yes	Male	5	P15	I think the system still at a **very primitive level**... {ummm...} [Deleted] and there are still **a lot of room to improve** the system..
-15	Bad	Negative	High	Yes	Female	5	P10	... I wanted to enjoy it but I was like ... ahhh shocked... [Deleted] ... I was taken aback... It wasn't that friendly I have to be frank with you. [Laughs] ...
-16	Neutral	Positive	Low	Yes	Male	4	P2	...it's not that a bad utility ... it's not that difficult to use. You still have to go aa... looking for things lah, and have a trial and error. ...
-19	Neutral	Positive	Low	No	Female	2	P9	("The spreadsheet is good... for umm... to replace... replace Excel-lah...") {("Why")} ("Because it... ah... how to say this... easier than Excel-lah.")
-19	Bad	Negative	High	Yes	Female	3	P11	(" [Pause] Difficult... because I am not familiar ...
-24	Bad	Negative	High	No	Female	4	P13	[Without hesitation] **I don't like it**. ...but this one is... it's easier but for someone who use to use Lotus it's difficult, you know? Okay?
-28	Bad	Positive	Low	No	Male	3	P8	*Ahhh... saya rasa ia lebih* (" I feel it is more")... effective if we use this.
-32	Bad	Negative	Low	Yes	Female	3	P1	Not user friendly.
-36	Neutral	Negative	High	No	Female	4	P5	("I think easy to use Excel.") [Laughs] ("Easier to use Excel.") {Oh} ("You.. you want to use, Excel is easier.")

Table 6.4: Aggregation of UAT data

Ordered by Aggregate Think Aloud Behaviour Count (TA Beh.). Aggregate Think Aloud Behaviour Count = Number of positive behaviours – Number of negative behaviours.

participants' data in the lower rows of the table. Data in the upper rows of the table would also indicate that the participants would have a better impression of the spreadsheet than participants whose data are located in the lower rows. Overall, high SUS ratings, which indicate that the system possesses good usability, were expected to appear in the upper rows. Similarly, positive interview comments are also expected to appear in the upper rows.

In Table 6.4, there are two types of UAT data. There are participants' data in which the UAT data agreed and data which did not agree. The following example illustrates a participant whose data agreed.

Participant P12 had relatively fewer problems as indicated by the low TA behaviour score that is fewer frustrations and negative comments as compared to other participants. This observation is supported by positive comments made in the think aloud session. For example, he noted the ease of entering the data:

> [P12:E70] Okay 'kay, *kita edit ya kita sunting* ("yes, we edit yes we edit") *S, Sunting.* ("S, Edit") This is easy. Electri-c- Electri-T-R-I-C- ... *betul* ("Correct?") Elec... Elec... Okay..

Participant P12 was also the only participant in the study who completed all the assigned tasks. Furthermore, participant P12 had the highest SUS ratings. He rated *Hamparan* 95 out of a possible 100. His good performance in the think aloud, and high SUS scores was reflected in his positive comment about the spreadsheet:

> [P12:Q1] *Bagus! Senang digunakan. Boleh komersilkan, saya rasa.* ("Good! Easy to use. Can be commercialised, I believe.")

However, there were some participants whose UAT data did not agree. In particular, the interview responses of some participants appear to disagree with the TA behaviour scores and the SUS data

There were only three positive behaviours compared to 31 negative behaviours observed in participant P8's think aloud data. In his negative comments, he expressed difficulty when using *Hamparan*:

> [P8:E44] Oh.. this one don't what you call it... the... centimeter? {Oh, okay}. Don't have ah [the unit]? Difficult to use lah. [Did not know unit of column width]

[P8:E70] Very difficult to use... whole day, [Laughs] make one file take whole day needs... [Laughs]

Participant P8's SUS score of 22.5 (out of 100) was the lowest SUS scores of all the participants. This SUS score means that P8 rated *Hamparan* as possessing very bad usability. However in the interview, P8 was unexpectedly positive about the spreadsheet:

[P8:Q1] *Ahhh... saya rasa ia lebih* ("I feel it [Hamparan] is more...") effective if we use this [spreadsheet]

In Table 6.4 there were four participants who gave positive comments despite having much more negative than positive behaviours. These participants, (P14, P2, P9 and P8 in shaded rows in Table 6.4), also did not rate the *Hamparan* as possessing positive usability in the SUS. However, their interview responses were quite positive and did not reflect the think aloud and SUS data.

Also, the interview response of the participant P3 (in the fourth row of Table 6.4) did not agree with the SUS and think aloud data. Given that this participant was ranked at the top half of the table, it would be expected that all the UAT data would reflect a positive perception of *Hamparan*. However, while the think aloud and SUS scores were positive, his interview response was quite negative.

Another participant with inconsistent data was participant P4. Despite her higher TA behaviour score and her positive comment, she did not rate *Hamparan* as positive in her SUS score. In fact her SUS score of 52.5 (see Table 5.4) was closer to a bad usability score (50) than a good usability score (60) see Table 6.4.

In light of the above observations, it would seem that not all the UAT data were consistent. Thus, the data from the UATs may not be able to determine whether participants had a good or bad perception of the software. As shown on Table 6.5, had all the UAT data agreed, the spreadsheet would have been identified as possessing either good or bad usability. However, given that not all the participants' data were consistent, the UAT tools were not able to determine tools' usability (unknown in Table 6.5). Thus, the aims of the usability evaluation were only partially achieved. The imported UATs could individually gauge the usability of the spreadsheet. Whether the individual measurement was plausible is not known until it is compared with measurements of other tools. However, when

the data collected by the UATs were examined together, the UATs appeared suspect given that all UAT data did not concur. This analysis is further discussed in the next chapter.

		Does software possess good usability?	
		Yes	No
Are all data from the UATs consistent?	Yes	Known	Known
	No	Unknown	Unknown

Table 6.5: Decision table of triangulation of UAT-data

Another irregularity was identified in the interview responses, but this time, with regards to the negative comments made. There were four participants who made quite negative comments: participant P3, P6, P15, and P13. These participants' negative comments would be expected to be located in the bottom half of the table rather in the top half. Negative comments would be associated with lower think aloud scores, that is more problems were experienced, and with more negative behaviours expressed by the participants. Furthermore, all these extreme negative remarks were made by higher status participants. It would seem that higher status participants were harsher in their comments compared to the low status participants' comments. As shown in Table 6.6, of the ten negative interview responses, eight were comments from high status participants. These comments made by high status participants, include: "I don't like it" [P13]; "very primitive" [P15]; "old fashioned" [P3]; "very outdated ... very basic" [P6]. Compared to the high status participants' comments, the negative comments made by the low status participants were less harsh, for example, "Not user friendly" [P1], "...it's not that a bad utility ... it's not that difficult to use. ..." [P2], "Umm... I think Excel *lebih senang lagi*... ("Excel is easier...") [P7]. On the other hand, of the seven positive comments expressed in the interview, five comments were from low status participants (P9, P8, P4, P2 and P14). The only high status participant with a positive comment was participant P12. However, P12's positive interview data concurred with the TA behaviour and with the SUS score.

	Positive Interview Responses	Negative Interview Responses
High Status	2	8
Low status	5	2

Table 6.6: Status of participants and interview responses

6.3 Summary

The data analysed showed that the UATs performed as expected in collecting data for the improvement of the spreadsheet. However, while the data collected from the UATs on its own could indicate the spreadsheet's usability, there was disagreement in UAT data at the participant level. Six participants UAT data did not agree. The next chapter provides a discussion of the findings from this chapter and suggests possible explanations for the anomalies found.

Chapter 7 Discussion

The key question to be answered in this thesis is: Are imported UATs appropriate for the usability evaluation, as part of the global SDLC, of localised software? To answer this question, three imported UATs were employed in the usability evaluation of an internationalised and localised spreadsheet, *Hamparan*. Target Malaysian users, in Malaysia, were recruited to evaluate *Hamparan*. These imported UATs are considered appropriate for the Malaysian context, if they can collect data which can be employed to:

a. Improve the usability of Hamparan

b. Determine the usability of Hamparan.

The ensuing sections provide a discussion of the analyses detailed in Chapter 6, in the context of current literature. Section 7.1 reports on a discussion of the data collected to improve usability of *Hamparan* while Section 7.2 provides a discussion in relation to the data collected employed to determine the usability of *Hamparan*.

7.1 Improving *Hamparan's* Usability: A Discussion

From the analyses in the previous chapter, the usability problems, isolated from the think aloud and interview data, were sufficiently detailed to be applied to rectify problem areas. By using the information about each usability problem, the problem could usually be corrected. Rectifying usability problems should improve the usability of the spreadsheet.

The usability problems were also grouped into categories derived from usability heuristics. This categorisation provides an abstract view of the group of usability problems. The usability heuristics, on which the categories are based, could then also be used to solve those problems. An example to illustrate this use of the heuristics is shown in Section 6.1.2. Given the analysis reported in Chapter 6, the first aim was achieved. Imported UATs can be employed in formative evaluation, that is, the imported UATs can collect data which can be employed to improve *Hamparan*.

7.1.1 Appropriateness of UATs to Collect Improvement Data

To provide further support for the suitability of the UATs for the formative evaluation, the results of this study are compared with results of usability studies reported in literature.

One of the findings of this study has been that the think aloud usability problems were quite specific and detailed, and highlighted specific problem areas of *Hamparan* (as shown in Section 5.2). Although the interview usability problems were not so specific, these interview usability problems could also be employed to rectify problem areas in the spreadsheet. The "specificity" of the think aloud usability problems is also reported in literature such as Sweeney and Dillon (1987). They stated that think aloud/verbal protocol analysis would be able to highlight specific areas of the interface that caused problems for users. Nielsen (1993) also commented that think aloud was employed to pinpoint concrete interface elements that may cause misunderstanding. He also noted that the faulty interface could be re-designed. In sum, the usability problems identified from the data collected in this study using the think aloud method were specific and detailed like those reported in reported works.

While the think aloud usability problems were quite detailed, the interview usability problems detected were described as more "general" or "broad". Similarly, Yamagishi and Azuma (1987) noted that the interview data addressed more "global or higher level issues than the concurrent/think aloud verbal reports". A possible reason why interview problems were more general is that, while the think aloud data contains details of the participants' every action in the spreadsheet, the participants' interview responses contain an aggregate of their whole experience in using *Hamparan*. An example of this response is recorded as interview usability problem "Numerous functions were unavailable". This suggestion was given probably because the participants were unable to find functions they wanted to use to complete the tasks. The participants were probably familiar with functions found in other spreadsheets.

Despite the contrast in detail of the think aloud and interview usability problems, it was found that there was some conformity between the think aloud and interview usability problems. This agreement provided evidence that think aloud and interviews were valid UATs to identify usability problems in the

spreadsheet. The data from these two tools cross-validated each other. This agreement was also comparable to results found in Yamagishi and Azuma (1987). In that study, they reported substantial agreement between the problem reports obtained from their verbal protocol analysis, and the comments obtained in interview. They also reported that this agreement showed that their data were "mutually consistent".

Another result obtained from the comparison of the think aloud and interview usability problems, was that more think aloud usability problems than interview usability problems were identified. In this study, 63 think aloud usability problems were detected whereas only 20 interview problems were obtained. This result is similar to findings from Henderson et al. (1995). They had compared four usability evaluation tools: logging, questionnaire, interview and verbal protocol analysis/think aloud. UATs were employed in the usability evaluation of three types of software: a word processor, a spreadsheet, and a database. Henderson et al. (1995) found that verbal protocol analysis identified the largest number of problem areas, followed by interviews, questionnaire, and logged data. The verbal protocol analysis was also identified as the best tool irrespective of the type of software evaluated. Thus, it is probable that the imported UATs employed in this study are also suitable in the usability evaluation of other types of software, not just spreadsheets.

Henderson et al. (1995) also found that verbal protocol analysis, when combined with other tools, was the dominant tool for detecting problem areas. Since this study's think aloud and logging tool were combined, more usability problems were isolated from the data collected using augmented-think aloud than the interview. This finding concurs with that found by Henderson et al. (1995).

7.1.2 Cost

While more information was obtained from the augmented-think aloud data than the interview data, much more effort went into the analysis of the think aloud data than the interview data. Although the time taken to analyse the think aloud and logged data was not recorded in this study, there is no doubt that much more effort was needed to analyse the think aloud and logged data. The transcription of the think aloud data alone took much longer than the transcription of the interview data since more raw data was collected than from the interview

session. The length of the participant's think aloud audio record ranged from 15 to 48 minutes, whereas all the interview sessions were less than 10 minutes. In addition to the think aloud data, the logged data of each participant had to be examined in tandem with the think aloud data. The extra effort needed for the think aloud data analysis is again supported by Yamagishi and Azuma (1987), who found that 71% of the total time of analysis was spent for the verbal protocol analysis. In comparison, the time required for interview analysis was 5%, logged data 8%, and questionnaire data 16%.

With regards to the first aim, the imported UATs can be employed to collect data to improve the usability of the localised spreadsheet. In particular, the think aloud data was sufficiently detailed to provide necessary information for the improvement of spreadsheet usability. Furthermore, the findings based on this study's data were comparable to findings of a number of other studies. The similarity of this study's result compared to results of other studies provides further support that the imported UAT tools (think aloud and interview) are appropriate for the formative evaluation process.

7.2 Participant's Perception of *Hamparan:* A Discussion

The second aim in relation to the key question was: Imported UATs can collect data, which can then be employed to determine the usability of *Hamparan.* The usability of the spreadsheet would also indicate the success or failure of processes employed in the development of *Hamparan.*

In this study, imported UATs were employed to collect data on the usability of *Hamparan.* Each UAT was able to determine the usability of *Hamparan* on its own. However, these UAT data in isolation were insufficient to reliably determine the usability of the *Hamparan* since it was not known if the individual UATs were effective or if the data collected was actually accurate. The usability of *Hamparan* could be explored through a comparison by triangulation of the three UATs' data. Furthermore, the actual usability of *Hamparan* could only be determined if each participant's data set was internally consistent. A data set comprised the think aloud data, SUS data, and interview data of one participant. From the analyses in Chapter 6, eleven of the 17 participants' data sets agreed. Two participants' consistent data sets showed that *Hamparan* had

good usability, and nine participants' data consistently indicated that the spreadsheet had bad usability.

Unexpectedly, one third of the data sets did not agree (refer Section 6.2.1). These six anomalous data sets failed to indicate the usability of *Hamparan*. This high rate of internal inconsistency is "wasteful" given the high costs of employing participants in usability evaluation as well as the associated costs in collecting and analysing the data.

It would be useful to identify ways to reduce the number of inconsistent data sets. Looking at the usability experiment and the context of the study, it would seem that one or more of the following may have contributed to the anomalous data:

- the data collected may be inaccurate;
- the UATs employed to collect the data may not be effective – problems may lie with the evaluator, the participants, how the tool was used; or
- the software being evaluated – different types of software may result in more or less consistent UAT data sets.

Although it is not possible to determine which of the three "causes" effected the anomalous data, a further literature search was conducted to identify possible explanations for the anomaly.

An inspection of the six inconsistent data sets reveals that there were four participants whose interview data did not agree with think aloud and SUS data. These four participants had difficulty in using *Hamparan* as exemplified by high number of counts of negative behaviours compared to positive behaviours. They also rated, in the SUS, *Hamparan* as possessing poor usability. However, despite the poor performance and the poor SUS rating of *Hamparan*, these participants interview responses were quite positive. It would seem that participants were inclined towards positive responses in the interview despite poor performance in the spreadsheet and a negative rating in the SUS.

Neuman (1994) warned that, when using interviews, there may "be a courtesy bias whereby there exists strong cultural norms to provide answers that the interviewees think the interviewer wants to hear" or interviewees may want to

hide anything unpleasant. Neuman's comment appeared to explain the anomalous data obtained in this study whereby the participants' commented positively about the spreadsheet in the interview, despite facing difficulties in the think aloud session, and giving poor SUS scores.

The participants' positive tendency was also observed by Herman (1996) in a software evaluation conducted in Singapore. A participant actually broke down and cried during the software evaluation session. However, that participant was very positive about the software in the post-test interview. Herman attributed this behaviour to the Eastern culture whereby it is "considered culturally unacceptable to criticise the designer directly or openly, as this may cause the designer to lose face" (Herman, 1996). Similar observations were also made in the usability evaluation of public information kiosks in Singapore (Lim and Usma, 1998). This "polite" behaviour ties in with observations made by Nakakoji (1994). If Japanese users said they liked a system, it is possible that the test users may be telling the truth or the users were too polite to make negative comments about the system (Nakakoji, 1994). Sacher (1998) observed that when working with Asian users, "politeness, formal attitudes and different conventions toward expressing personal views can have significant impact on usability studies" (Sacher, 1998).

From the studies above, it would appear that the tendency of Asians to be polite or to hide any "unpleasantness" might have contributed to the positive comments in the interview despite poor performance and poor SUS ratings. While this politeness behaviour in the interview may explain the positive comments, it does not explain the harsh negative comments that were provided by other participants. These positive and negative comments expressed in the interview, however, may be explained by power distance.

Power distance is defined as "the extent to which the less powerful members of institutions and organisations within a country expect and accept that power is distributed unequally." (Hofstede, 1994). From a sample of 50 countries and three regions, Hofstede identified Malaysia as the country having the highest power distance. Singapore is ranked 13th, Indonesia 8/9th, Thailand 21/23rd, US 40th, Australia 36th, Britain 42/44th, New Zealand 50th (see Table 7.1).

Score Rank	Country or region	PDI Score	Score Rank	Country or region	PDI Score
1	Malaysia	104	27/28	South Korea	60
2/3	Guatemala	95	29/30	Iran	58
2/3	Panama	95	29/30	Taiwan	58
4	Philippines	94	31	Spain	57
5/6	Mexico	81	32	Pakistan	55
5/6	Venezuela	81	33	Japan	54
7	Arab countries	80	34	Italy	50
8/9	Equador	78	35/36	Argentina	49
8/9	Indonesia	78	35/36	South Africa	49
10/11	India	77	37	Jamaica	45
10/11	West Africa	77	38	USA	40
12	Yugoslavia	76	39	Canada	39
13	Singapore	74	40	Netherlands	38
14	Brazil	69	41	Australia	36
15/16	France	68	42/44	Costa Rica	35
15/16	Hong Kong	68	42/44	Germany FR	35
17	Colombia	67	42/44	Great Britain	35
18/19	Salvador	66	45	Switzerland	34
18/19	Turkey	66	46	Finland	33
20	Belgium	65	47/48	Norway	31
21/23	East Africa	64	47/48	Sweden	31
21/23	Peru	64	49	Ireland (Republic of)	28
21/23	Thailand	64	50	New Zealand	22
24/25	Chile	63	51	Denmark	18
24/25	Portugal	63	52	Israel	13
26	Uruguay	61	53	Austria	11
27/28	Greece	60			

Table 7.1: Power distance index (PDI) values (sourced from Hofstede (1994))

This ranking means that Malaysians in general are willing to accept the fact of inequality in power as being normal. Hofstede stated that subordinates in high power distance countries, are unlikely to contradict their bosses directly. The power holder's authority is unquestioned.

This viewpoint is supported by Asma Abdullah (1996) who observed that in Malaysia, subordinates are unlikely to openly reject the viewpoints of their supervisors. Subordinates who do question a power holder's authority would be seen as behaving improperly and disrespectfully. Furthermore, in a high power distance country such as Malaysia, employees are "afraid" of their employers as employers wield powers such as the authority to fire employees. Thus, a person of higher status and power, such as a manager, will be more likely to voice his or her feelings of discontent to a person of lower status, for example, a subordinate. However, the reverse is not true. A person of lower status is unlikely to go against a person of higher status for fear of retribution.

The four anomalous results of this study consisting of positive interview responses of P14, P2, P9, and P8 can be explained with respect to this power distance characteristic. All these four participants were low status participants. Given the number of expressions of frustrations and negative comments made by

the lower status participants in the think aloud session, these participants probably did not like *Hamparan* any more than the higher status participants. However, these lower status participants were more positive about *Hamparan* as they did not want to question or "go against" the experimenter – a tutor, considered a high status/power holder – maybe for fear of retribution or of appearing disrespectful to the experimenter. The low status participants thus were less critical and "less honest" in their responses, to the extent of suggesting to replace Excel with the DOS spreadsheet [participant P9]. On the other hand, high status participants were more likely to voice their dissatisfaction as they were of the same or higher status (compared to the experimenter) in the organisation hierarchy, and they had little fear of retribution. This observation is supported by the frank comments made by the higher status participants who performed poorly in using the spreadsheet, but were more forthright in their comments. One high status participant was quite candid. When asked her opinion of the spreadsheet, she unhesitatingly and bluntly replied "I don't like it [the spreadsheet]" [P13]. Comparing all the negative interview responses, the higher status participants' responses were also harsher than comments of lower status participants. Some of the negative comments contained intensifiers such as "very", "quite" and "really", for example "very primitive" [P15] and "very outdated" and "very basic" [P6].

This power distance explanation was reported and presented at the Annual Conference of the UK Ergonomics Society (see Yeo, Barbour and Apperley (1998)) and at the CHI 98 Doctoral Consortium (refer Yeo (1998)). Conference attendees, who had worked in countries with high power distance, provided positive feedback. This feedback gave further support to the power distance explanation.

In the remaining two anomalous results, it was the SUS data that did not agree with the other two UAT data. The anomalous results of the two remaining participants, P3 and P4, may be explained by the politeness and status-disparity attribute of the participants. With participant P3 (a high status participant) despite having problems using the spreadsheet, he may not have liked the spreadsheet, and said so in the interview; he called *Hamparan* "old fashioned". However, in the SUS, he may have wanted to save the feelings of the experimenter, thus gave a positive rating of *Hamparan* in the SUS. With regards to participant P4, she appeared to be able to complete almost all the tasks with relatively few

difficulties, she may have had a favourable opinion of the spreadsheet, and this was substantiated by her positive interview response. Despite her relatively positive comment (she said it was better to use the software in her department) her SUS rating of 52.5 (neutral rating) may have revealed her true thoughts about the spreadsheet.

Initially, it appeared that only six (35%) of the 17 participants' data were inconsistent. However, if the status attributes are considered, only one (10%) of the ten high status participants' data sets was anomalous. Five (71%) of the seven low status participants' data did not agree. This finding suggests that more consistent data may be obtained if high status participants were employed. If an experiment in Malaysia employed low status participants and only interviews were used to collect the data, the results may not be accurate. In addition, inaccurate usability data would in turn affect the quality of decisions made about the spreadsheet. For example, a software application may be released prematurely due to participants' very positive responses, only to find out afterwards that the participants did not like the spreadsheet but they did not want to say so to the company vice-president who conducted the interviews.

While these participants' characteristics may to a certain extent explain the observations, this investigation of the data sets lacks rigour since the examination was mainly based on an inspection of the data sets. As such, a more rigorous re-examination of the UAT data is warranted. This re-examination may identify fuller explanations of the observations presented thus far.

7.3 Summary

The key question to be answered in this thesis is: Are imported UATs appropriate for the usability evaluation, as part of the global SDLC, of the localised software? The results of the study indicate that imported UATs are not totally appropriate in the usability evaluation of a localised tool. While the UATs can be employed to collect data, which can be applied to improve the spreadsheet's usability, the imported UATs appear to be only partially effective in ascertaining the usability of *Hamparan*. It would appear, from initial investigation of the data, that the status-disparity of the participants might have contributed to the anomalies of the UAT results. A re-examination of the UATs has to be

conducted to determine whether the status-disparity attribute may have actually contributed to the data inconsistency. The next chapter details procedures as well as results of the re-examination of the UAT data sets.

Chapter 8 Re-Examination of Usability Data

The key question to be answered in this thesis is: Are imported usability assessment tools (UATs) appropriate for the usability evaluation, as part of the global-SDLC (global-software development life-cycle) of localised software? To answer this question, data was collected employing three UATs as reported in Chapter 6 and 7. The data collected could be employed to improve the usability of *Hamparan.* However, there were some participants whose data sets were anomalous. The usability of *Hamparan* could only be correctly determined by using consistent data sets. This chapter describes a re-examination of the data in an attempt to account for the anomalies.

8.1 Introduction

To answer the key question above, one aim of the usability evaluation was to determine whether the imported UATs could collect data which could be employed to determine usability of *Hamparan.* Since three UATs were employed, only when data sets were consistent could the usability of *Hamparan* be determined. As reported in Chapter 7, there were a number of inconsistent data sets (anomalous data) that were detected. Six of the 17 participants' data sets were inconsistent. When participants' status was considered, only one (10%) of the ten high status participants' data sets was anomalous while five (71%) of the seven low status participants' data were not consistent.

Thus, the objective of this re-examination of data was to identify which participants' characteristics may have contributed to the consistency of the UAT data sets. One conjecture would be: High status participants' data sets are more consistent than low status participants' data sets. Thus, if it was found that data sets of high status participants were more consistent than low status participants, the UATs may need to be adapted (depending on the participants employed) so that more consistent data are collected. Rather than just limiting the characteristics to just status, other participants' attributes not considered in this study were also examined. These other characteristics examined are detailed in the next section.

8.2 Method

This section describes the approach taken to ascertain the consistency of the UAT data sets. The data collected using the three UATs were compared to check for agreement and consistency in the following manner.

UAT Rankings

Rankings were derived and compared for each of UAT data sets and the rankings were compared. To obtain the ranking from the think-aloud (TA) objective-measures, an overall TA behaviour score for each participant was derived. Henceforth, the TA behaviour score is referred to as aggregate TA score. The TA score was derived by subtracting the total number of observed negative behaviours from the total number of observed positive behaviours. The think aloud totals were then sorted in descending order and ranked from 1 (highest score), to 17 (lowest score). A low TA score or rank would indicate that the participant had relatively more negative than positive behaviours compared to a participant with a higher TA rank. These scores also reflect the actual performance of the participants during the think aloud session. A low rank would suggest the participants performed poorly, the participant encountered relatively more problems since more frustrations and negative comments were expressed compared to a participant with a higher rank.

The 17 SUS scores were also ranked from 1 to 17; rank 1 was assigned to the participant with the highest SUS score, and rank 17, the lowest SUS score. As such, participants who had high SUS scores (high SUS rankings such as rank 1) would have a more positive opinion of the spreadsheet than participants who had rated the spreadsheet poorly (lower SUS rankings, such as rank 17).

The participants' opinions of *Hamparan* collected from the interview were ranked in the following manner. Three evaluators were recruited to rank the interview responses. These three evaluators were Malaysians literate and fluent in both Bahasa Melayu and English. They would be able to detect any nuances of the interview responses. The evaluators ranked, in their opinion, the seventeen interview responses from the most positive interview response, as rank 1, to the most negative interview response, as rank 17. Each participant's three interview rankings, given by the three evaluators, were totalled and these totals ranked/ordered again. Thus, the interview responses with the lowest total would

be ranked 1, that is, the most positive opinion expressed. The highest total would be ranked 17, which is the most negative opinion articulated. If there were tied scores, the ranks would be averaged as described in Levin and Fox (1997). For example, if the participants in ranks 3 and 4 had tied scores, an average rank of 3.5, that is, (3+4)/2, would be assigned to each participant. Tied aggregate TA scores and SUS scores were also dealt with in this manner.

Rankings of the scores were used to determine the consistency since it was difficult to provide a quantitative measurement to the interview responses. A qualitative evaluation of the responses was more appropriate to provide an indication of the participants' attitude towards *Hamparan*. Thus, the interview responses were ranked. Since the interview responses were ranked, the other UAT data were then converted to ranks (Heiman, 1992) so that a comparison of the various UATs could be conducted.

Spearman's Ranked Correlation Coefficient

Once the rankings of all three UATs had been determined, the Spearman's Ranked Correlation Coefficient, henceforth known as correlation coefficient, was calculated to determine how well or consistently the ranking of one UAT correlated with the ranking of another UAT. Nielsen and Levy (1994) indicated that objective and subjective measures correlated positively. An assumption is made that highly correlated variables will have the same rank order, that is, the rankings from TA, SUS and interview responses will all have the same rank order. The correlation coefficients were calculated for each of the three combination pairs of UAT rankings namely, the TA-SUS pair, the TA-interview pair, and lastly, SUS-interview pair. All these combination pairs are collectively known as UAT-pairs. Highly correlated UAT-pairs that have exactly the same rank order will have a correlation coefficient of +1, a perfect positive correlation. For example, a strong positive correlation coefficient in the TA-SUS pair would suggest that order of TA rankings is very similar to the order of SUS ranking. In addition to the correlation coefficients of the UAT-pairs, the correlation coefficients of interview rankings of the three evaluators employed were also calculated to determine the consistency of these evaluators' ranking.

According to Levin and Fox (1997), correlation coefficients of about +0.6 are denoted as strong positive correlations, +0.3 denoted as moderate positive

correlations, and +0.1, weak positive correlation. Likewise, correlation coefficients of about -0.6 are denoted as strong negative correlations, -0.3 denoted as moderate negative correlations, and -0.1, weak negative correlation. Levin and Fox (1997) also state that the strength of the correlations is independent of direction, thus, coefficients of -0.8 and +0.8 are of equal strength, both are very strong correlations.

Besides the correlation coefficients, p-values of the correlation coefficients were also calculated. According to Guyatt et al. (1995), this *p* value is ascertained from a hypothesis test, with the null hypothesis being that the true correlation between the two rankings is 0; that is, there is no correlation between the two rankings. The *p* value represents the probability that, if the true correlation were 0, a relation as strong as or stronger than the correlation actually observed would have occurred by chance. The smaller the *p* value, the less likely it is that chance explains the apparent relation between the two measures (Guyatt et al., 1995). The conventional threshold for statistical significance is 0.05. If the p-value is less than 0.05, the alternative hypothesis, that the correlation is not 0, is accepted.

Sub-groups

The correlation coefficients of UAT-pairs of rankings of all 17 participants were calculated. In addition to the status characteristic, correlation coefficients of the rankings of other sub-groups of participants were also calculated. These sub-groups were derived according to the participants' profiles. The sub-groups were mutually exclusive sub-groups; one sub-group has the attribute, the other does not. For example, one such profile breakdown was Lotus 123 (DOS) experience, whereby two sub-groups were obtained: participants with Lotus 123 (DOS) experience, and participants without Lotus 123 (DOS) experience. A participant with Lotus 123 (DOS) experience meant that the participant had used the DOS version of Lotus 123 prior to the experiment. The other attributes used for the profile breakdown include: the familiarity of participants with the experimenter; the participants' DOS experience; number of years the participants had used a computer; the participants' total number hours of spreadsheet-use; the participants' gender and their age. A person is said to be familiar with the experimenter if the participant has had at least one ten-minute

one-on-one conversation with the experimenter. Correlation coefficients of the participants' rankings are presented in the next section.

8.3 Results

Table 8.1 shows the rankings of the interview responses provided by each of the three evaluators. All three evaluators ranked the same participants for ranks 1, 2, 3, 4, and 17. The correlation coefficients of the evaluator pairs of rankings are presented in Table 8.2. The correlation coefficients show very strong positive correlation. Also, the correlation coefficients of the evaluator's rankings associated with the final interview rankings used (based on the total of rankings of the three evaluators) were better than the correlation coefficients of the pairings of evaluators' rankings. All correlation coefficients were statistically significant at 5% significance level. Table 8.3 contains the UAT rankings of all the participants. The profile of each participant is also included in the table.

Partici-pant	Evaluator 1	Evaluator 2	Evaluator 3	Total of 3 ranks	Interview rank
P12	1	1	1	3	1
P9	2	2	2	6	2
P8	3	3	3	9	3
P4	4	4	4	12	4
P14	6	5	7	18	5
P7	5	10	5	20	6
P16	8	6	11	25	7
P5	7	9	10	26	8
P17	13	8	8	29	9
P15	11	7	12	30	10
P3	10	15	6	31	11
P2	12	13	9	34	12.5
P6	9	12	13	34	12.5
P10	15	11	14	40	14
P1	14	16	15	45	15
P11	16	14	16	46	16
P13	17	17	17	51	17

Table 8.1: The three evaluator's rank order of interview responses

	E1-E2	E1-E3	E2-E3
Correlations of rankings of evaluators	0.836 (0.000)	0.892 (0.000)	0.765 (0.000)
	FinalRk-E1	**FinalRk-E2**	**FinalRk-E3**
Correlations between rankings of evaluators and the final interview ranking	0.955 (0.000)	0.928 (0.000)	0.917 (0.000)

Table 8.2: Correlation coefficients of rankings of evaluators and final ranking
p-values are in brackets

	Rankings			Absolute Differences of ranks				Participants' Profiles							
Participants	Tar (Tr)[1]	SUSr (Sr)[2]	IntR (Ir)[3]	Tr-Sr	Tr-Ir	Sr-Ir	Totals	Fam[a]	Lotus 123[b]	DOS	Yrs. Used	Hrs. Wk	Status	Age	Gender
P8*	15	17	3	2	12	14	**28**	**No**	**No**	Yes	< 5	> 40	Low	≤ 26	Male
P5	17	4	8	13	9	4	**26**	**No**	**No**	No	< 5	≤ 40	High	27-36	Fem.
P17	5	16	9	11	4	7	**22**	**No**	**No**	No	> 9	≤ 40	High	≤ 26	Male
P9*	12.5	6.5	2	6	10.5	4.5	**21**	**No**	**No**	No	< 5	> 40	Low	27-36	Fem.
P3*	3.5	2	11	1.5	7.5	9	18	Yes	Yes	Yes	> 9	> 40	High	27-36	Male
P16	1	3	7	2	6	4	12	Yes	No	Yes	> 9	> 40	High	≤ 26	Fem.
P2*	11	6.5	12.5	4.5	1.5	6	12	Yes	Yes	Yes	> 9	> 40	Low	27-36	Male
P6	7	10	12.5	3	5.5	2.5	11	No	Yes	Yes	> 9	> 40	High	27-36	Male
P14*	8.5	10	5	1.5	3.5	5	10	No	No	No	< 5	≤ 40	Low	≤ 26	Male
P15	8.5	13.5	10	5	1.5	3.5	10	Yes	Yes	Yes	> 9	> 40	High	27-36	Male
P1	16	12	15	4	1	3	8	Yes	No	No	< 5	> 40	Low	27-36	Fem.
P10	10	10	14	0	4	4	8	Yes	Yes	Yes	> 9	> 40	High	27-36	Fem.
P13	14	13.5	17	0.5	3	3.5	7	No	Yes	Yes	> 9	> 40	High	≤ 26	Fem.
P11	12.5	15	16	2.5	3.5	1	7	Yes	Yes	Yes	> 9	> 40	High	27-36	Fem.
P4*	3.5	6.5	4	3	0.5	2.5	6	No	Yes	Yes	< 5	> 40	Low	≤ 26	Fem.
P12	2	1	1	1	1	0	2	Yes	Yes	Yes	> 9	> 40	High	≤ 26	Male
P7	6	6.5	6	0.5	0	0.5	1	Yes	No	No	< 5	≤ 40	Low	≤ 26	Fem.

Table 8.3: Rankings and profile of participants
*[1] Think aloud rank; [2] SUS rank; [3] Interview rank; [a] Familiarity of participants with the experimenter; [b] Lotus 123 Experience of participants; * Participants identified with anomalous data sets in Chapter 6.*

The three combination UAT-pairs (of rankings) of all 17 participants showed moderate positive correlation, see Figure 8.1 with label "All participants". In particular, the TA-SUS pair showed the strongest positive correlation of the three UAT-pairs. In the sub-group of participants with Lotus 123 (DOS) experience, all three correlation coefficients of UAT-pairs of rankings, were very strong positive correlations from 0.70 to 0.90, refer Figure 8.1 (Lotus 123: Yes). All these coefficients were statistically significant at 5% significance level. The values of the correlation coefficients and their respective p-values are available in Appendix K. In contrast, all three correlation coefficients of UAT-pairs (of rankings) of participants without Lotus 123 (DOS) experience were weak, 0.01 to 0.17, see Figure 8.1 (Lotus-123: No). These coefficients indicate that rankings of UAT-pairs correlated poorly.

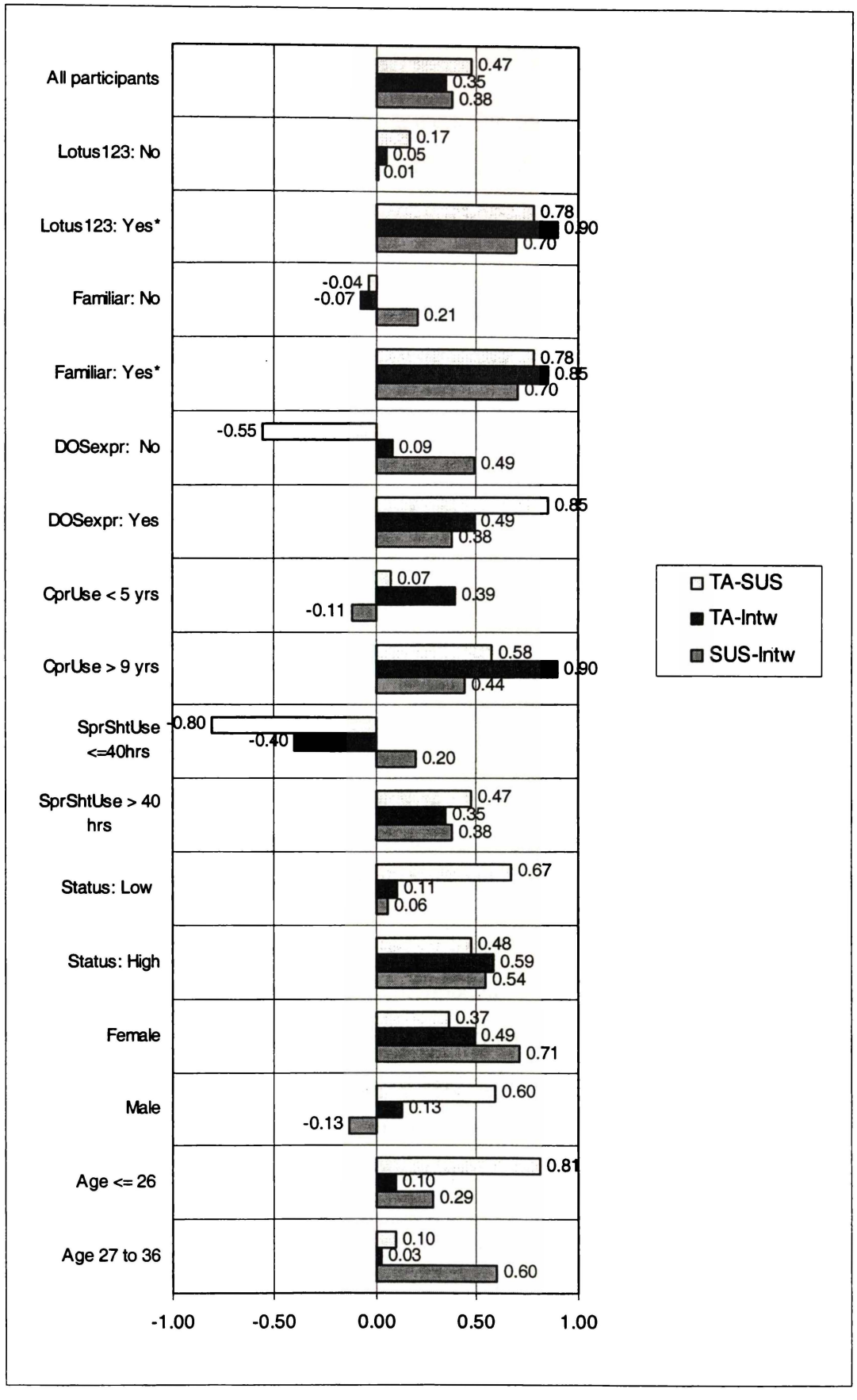

Figure 8.1: Spearman's Ranked Correlation Coefficients of UAT-pairs
**All correlations were statistically significant at 5% significance level.*
See Appendix K for p-values; Intw: Interview; TA-SUS: TA and SUS pair.

Another interesting observation was that all correlation coefficients of UAT-pairs of:

- participants who used computers for more than 9 years, were greater than correlation coefficients of participants who used computers 4 years or less, refer Figure 8.1 (CprUse > 9 yrs and CprUse < 5 yrs)
- participants with DOS experience, were greater than the correlation coefficients of participants without DOS experience except the SUS-interview pair, see Figure 8.1 (DOSexpr: Yes and DOSexpr: No) and
- participants who had more than 40 hours of spreadsheet experience, were greater than correlation coefficients of participants who had at most 40 hours of spreadsheet experience, see Figure 8.1 (SprShtUse >40 hrs and SprShtUse <=40hrs,).

On the whole, the more-experienced computer-users had moderate to strong positive correlation coefficients (0.35 to 0.90) compared to their less experienced counterparts who had strong negative to moderate positive correlation coefficients, from -0.80 to 0.39. Refer to Figure 8.1 (Lotus123: Yes*, DOSexpr: Yes, CprUse > 9 yrs, SprShtUse > 40 hrs) for all correlation coefficient values.

Another important contrast was observed in sub-groups pertaining to participants' familiarity with the experimenter. UAT-pairs (of rankings) of participants who were familiar with the experimenter had strong positive correlation; refer Figure 8.1 (Familiar: Yes). All coefficients were statistically significant at 5% significance level. In contrast, rankings of participants who were not familiar with the experimenter did not correlate well; in fact, two of the three UAT-combinations had negative correlations, see Figure 8.1 (Familiar: No).

Although the correlation coefficients of rankings of the status sub-groups were not statistically significant at 5% significance level, all the correlation coefficients of high status participants were moderate to strong positive correlation, from 0.48 to 0.59, see Figure 8.1 (Status: High). Only the correlation coefficients of the TA-SUS pair of low status participants had strong positive correlation, that is, 0.67. The correlation coefficients of low status participants'

TA-Interview and SUS-Interview pairs were weak positive correlation, 0.11 and 0.06 only. These correlation coefficients indicate rankings of the interview correlated poorly with rankings of both TA and SUS.

8.4 Analysis

The evaluators' ranking of the interview responses were highly correlated. This high correlation indicates that the ranks provided agreed. With respect to the correlation coefficients of rankings of all 17 participants, all the three combination pairs of UAT-rankings were moderate positive correlations. This result supports the assumption that the UAT-pairs of rankings would positively correlate, that is the rank order of one UAT would match the rank order of the other UAT. However, not all UAT-pairs of rankings of sub-groups correlated as expected.

All the three UAT-pairs of rankings of the sub-groups, "participants with Lotus 123 (DOS) experience" and "participants familiar with the experimenter", were statistically correlated at 5% significance level. Thus, there is evidence to indicate that in these sub-groups, higher TA ranks would also indicate higher SUS and interview rankings, and vice versa. In other words, the TA behaviour/performance measure was consistent/agreed with the subjective SUS scores and interview responses; a participant in these sub-groups with a low rank in the TA would also have a low rank in the SUS and interview, similarly for high rank. As such, participants who performed poorly in the TA session would have rated the spreadsheet as having poor usability and those participants would also have made more negative comments.

In contrast, in sub-groups of "participants without Lotus 123 (DOS) experience" and "participants who were not familiar with the experimenter", the UAT ranks did not correlate, that is, the responses in the UATs were not consistent. Participants in these sub-groups with low TA ranks may not necessarily have low rank in the SUS or interview. Thus, a participant who had performed poorly in the TA made more negative comments or expressed more frustrations. The participants may rate the spreadsheet in the SUS as possessing good usability or may even provide positive comments in the interview, as noted in the anomalous data in Section 6.2.1.

Given that UAT-pairs of rankings of certain sub-groups correlated better, further analysis was conducted to isolate participants whose UAT rankings may have contributed to the inconsistencies/poor correlation. To do this, the absolute differences of each of the three combination pairs of UAT rankings were aggregated (see Totals column in Table 8.3). Higher Totals of absolute differences indicated a greater "variability" or "inconsistency" in all the UAT-pairs, that is, the UAT-pairs indicate weaker correlations. Lower Totals would suggest a stronger correlation among the three UAT-pairs. In Table 8.3, the Totals column was sorted in descending order of combined absolute-differences.

From the Table 8.3, all the four participants (P8, P5, P17 and P9) with the highest absolute difference Total were not experienced in Lotus 123 (DOS) and were not familiar with the experimenter. These participants' high inconsistencies probably contributed to the poor correlation of UAT responses in the subgroups "participants without no Lotus 123 (DOS) experience". This observation concurs with the poor correlation coefficients of the UAT-pairs (of rankings) in those two sub-groups. Note also that all nine participants, who had Lotus 123 (DOS) experience were also participants with experience in DOS, used computers more than nine years (except participant P4), and had used spreadsheets more than 40 hours.

Although the status sub-group's results were not statistically significant, an analysis of the interview responses supports the observations made in Chapter 7 that high status participants provided more harsh negative comments than low status participants. Table 8.4 presents the participants' interview ranking ordered in ascending order. (Details of low status participants are in shaded rows in Table 8.4.) Rank 1 is the most positive interview response, which was made by participant P12, and rank 17 the most negative response. With the exception of participant P12 (who performed well in the TA session, rated *Hamparan* with the highest SUS scores, and was enthusiastic about the software), interview ranks 2 to 6 were responses from low status participants. Also, note that the remaining two participants P1 and P2 who gave negative comments, but both were participants who were familiar with the experimenter. The significance of this observation will be revealed in the next section.

	Rankings			Absolute Differences of ranks				Participants' Profiles							
Participants	TAr (Tr)[1]	SUSr (Sr)[2]	IntR (Ir)[3]	Tr-Sr	Tr-Ir	Sr-Ir	Totals	Familiarity[a]	Lotus 123[b]	DOS	Yrs. Used	Hrs. Wk	Status	Age	Gender
P12	2	1	1	1	1	0	2	Yes	Yes	Yes	> 9	> 40	High	≤ 26	Male
P9*	12.5	6.5	2	6	10.5	4.5	21	No	No	No	< 5	> 40	Low	27-36	Fem.
P8*	15	17	3	2	12	14	28	No	No	Yes	< 5	> 40	Low	≤ 26	Male
P4*	3.5	6.5	4	3	0.5	2.5	6	No	Yes	Yes	< 5	> 40	Low	≤ 26	Fem.
P14*	8.5	10	5	1.5	3.5	5	10	No	No	No	< 5	≤ 40	Low	≤ 26	Male
P7	6	6.5	6	0.5	0	0.5	1	Yes	No	No	< 5	≤ 40	Low	≤ 26	Fem.
P16	1	3	7	2	6	4	12	Yes	No	Yes	> 9	> 40	High	≤ 26	Fem.
P5	17	4	8	13	9	4	26	No	No	No	< 5	≤ 40	High	27-36	Fem.
P17	5	16	9	11	4	7	22	No	No	No	> 9	≤ 40	High	≤ 26	Male
P15	8.5	13.5	10	5	1.5	3.5	10	Yes	Yes	Yes	> 9	> 40	High	27-36	Male
P3*	3.5	2	11	1.5	7.5	9	18	Yes	Yes	Yes	> 9	> 40	High	27-36	Male
P6	7	10	12.5	3	5.5	2.5	11	No	Yes	Yes	> 9	> 40	High	27-36	Male
P2*	11	6.5	12.5	4.5	1.5	6	12	Yes	Yes	Yes	> 9	> 40	Low	27-36	Male
P10	10	10	14	0	4	4	8	Yes	Yes	Yes	> 9	> 40	High	27-36	Fem.
P1	16	12	15	4	1	3	8	Yes	No	No	< 5	> 40	Low	27-36	Fem.
P11	12.5	15	16	2.5	3.5	1	7	Yes	Yes	Yes	> 9	> 40	High	27-36	Fem.
P13	14	13.5	17	0.5	3	3.5	7	No	Yes	Yes	> 9	> 40	High	≤ 26	Fem.

Table 8.4: Participants' ranks sorted by interview rank
*[1] Think aloud rank; [2] SUS rank; [3] Interview rank; [a] Familiarity of participants with the experimenter; [b] Lotus 123 Experience of participants; * Participants identified with anomalous data sets in Chapter 6. Shaded rows are low status participants.*

8.5 Discussion

In the introduction to this chapter, it was speculated that the data sets of high status participants were more consistent than data sets of low status participants. Only if data sets from UATs are consistent, can the usability of *Hamparan* be ascertained with sufficient confidence. From the analysis of results, it would appear the UAT-pairs of rankings of high status participants correlated better than the lower status participants' UAT-pairs of rankings. While the high status participants' correlation coefficients were not statistically significant at 5% significance level, there were data sets of other sub-groups which provided more consistent results.

The UAT data sets appeared to be more consistent when specific sub-groups of participants were used. UAT data sets of participants with Lotus 123 (DOS) experience and participants familiar with experimenter were found to be consistent, based on evidence of the correlation coefficients of UAT-pair rankings. This finding means that a high rank in one UAT data is more likely to indicate a high rank in other UATs' data. Data sets from these participants are more likely

to be more consistent and thus, can be more reliably employed to determine the usability of *Hamparan.* However, the UATs' rankings were found to be inconsistent when participants without Lotus 123 (DOS) experience and participants unfamiliar with the experimenter were used. This inconsistency was identified based on the correlation coefficients as well as the aggregate absolute-differences of UAT-pair ranks, refer Table 8.3. Possible explanations of the consistencies and inconsistencies of the data sets are provided in the following sections.

8.5.1 Computer Experience

Kissel (1995) found that objective measures (the time taken to complete tasks using different interfaces) and subjective measures (survey of preference of interface, ease of use) did not necessarily match. However, objective measures and subjective measures of participants' who were experienced computer-users, matched (Kissel, 1995). Experienced users in Kissel's (1995) study were subjects who were familiar with a variety of types of software, and used computers on a regular basis.

Similarly in this study, the correlations of UAT-pairs of rankings of participants who had Lotus 123 (DOS) experience were statistically significant, as compared to participants without Lotus 123 (DOS) experience, which were not. Sub-groups based on participants' familiarity with DOS, participants who had used computers at least 9 years, as well as participants who had used spreadsheets more than 40 hours, also had better correlations of UAT-pairs than their less experienced counterparts.

Kissel (1995) did not furnish reasons as to why the objective and subjective measures corresponded when using participants with more computer experience. The following explanation may be able to account for the strong correlations in this study. All participants who had Lotus 123 (DOS) experience were also knowledgeable in DOS and had more than nine years of computer experience (with the exception of participant P4 who had only 3 years' computer experience). Thus, all these participants were quite familiar with spreadsheets and computers given their past experience. These participants had problems using *Hamparan,* a DOS spreadsheet, obviously less sophisticated than the Windows version of Microsoft Excel with which they were familiar. The participants were

thus confident of their "opinions" and thus were candid in their comments, especially in the interview. Nielsen and Levy's (1994) noted that "the experienced users tend to be more extreme in their relative evaluations of the systems being compared".

In contrast, participants such as P8 and P9 (see Table 8.3), who were not experienced in Lotus 123 (DOS), were positive about *Hamparan* in their interview responses despite poor performance in the TA sessions. Their interview rankings were higher than their TA rankings. These participants probably felt that they were not qualified, possibly lacking in confidence, to give comments about the spreadsheets since they were not as knowledgeable about computers and spreadsheet. Thus, these participants may have opted for positive comments despite experiencing more difficulties using the spreadsheet, as indicated by their low TA ranks.

Score Rank	*Country or region*	*IDV Score*	*Score Rank*	*Country or region*	*IDV Score*
1	USA	91	28	Turkey	37
2	Australia	90	29	Uruguay	36
3	Great Britain	89	30	Greece	35
4/5	Canada	80	31	Philippines	32
4/5	Netherlands	80	32	Mexico	30
6	New Zealand	79	33/35	East Africa	27
7	Italy	76	33/35	Yugoslavia	27
8	Belgium	75	33/35	Portugal	27
9	Denmark	74	36	Malaysia	26
10/11	Sweden	71	37	Hong Kong	25
10/11	France	71	38	Chile	23
12	Ireland (Rep. of)	70	39/41	West Africa	20
13	Norway	69	39/41	Singapore	20
14	Switzerland	68	39/41	Thailand	20
15	Germany F.R.	67	42	Salvador	19
16	South Africa	65	43	South Korea	18
17	Finland	63	44	Taiwan	17
18	Austria	55	45	Peru	16
19	Israel	54	46	Costa Rica	15
20	Spain	51	47/48	Pakistan	14
21	India	48	47/48	Indonesia	14
22/23	Japan	46	49	Colombia	13
22/23	Argentina	46	50	Venezuela	12
24	Iran	41	51	Panama	11
25	Jamaica	39	52	Equador	8
26/27	Brazil	38	53	Guatemala	6
26/27	Arab Countries	38			

Table 8.5: Individualism index (IDV) values (sourced from Hofstede (1994))
Rank 1 is most individualistic country; Rank 53, most collectivistic country.

8.5.2 Collectivism

According to Hofstede (1994), Malaysia is a more collectivistic than an individualistic country. Malaysia is ranked 36 out of 50 countries and 3 regions in

an individualist-collectivist continuum. At the individualist end, US is ranked 1st, Australia 2nd, Great Britain 3rd, New Zealand 6th, while at the collectivist end, Hong Kong is ranked 37th, and Singapore 39th (see Table 8.5). Malaysia's collectivist ranking means that Malaysians prefer a closely-knit social framework as opposed to a loosely-knit social framework (Zabid, Anantharaman and Raveendran, 1997). Also, the concerns of a group are more important than the concerns of an individual (Hofstede, 1994) in Malaysia. Thus, Malaysians have a high concern for others, keeping other people in mind, advocating a sense of unity with other people, and considering the group as a basic unit of survival (Asma Abdullah, 1996).

Two values that are important to a collectivist society are maintenance of harmony and the preservation of face (Hofstede, 1994). Values here are defined as a set of clear and uncompromising statements, what are important to a specific group (Asma Abdullah, 1996). These values influence how a person of a particular cultural group think, feel, and act, as well as how a person works with others (Asma Abdullah, 1996). The maintenance of harmony is important to perpetuate the closely-knit social framework. One way of maintaining harmony is the preservation of face. Preserving face means maintaining one's dignity by not embarrassing or humiliating a person in front of others (Zabid, Anantharaman, & Raveendran, 1997). By preserving one's face, interpersonal relations can be improved, and harmony as well as respect can be sustained (Zabid, Anantharaman, & Raveendran, 1997). Also, direct confrontation of another person is considered rude and undesirable in most collectivist cultures (Hofstede, 1994). In Malaysia, giving frank negative opinions can therefore undermine harmonious relationships and threaten group solidarity (Asma Abdullah, 1996). This preservation of harmony is exemplified in a quotation from Malaysia's former Foreign Minister, Datuk Seri Abdullah Haji Ahmad Badawi, on the need to preserve the Association of South-East Asian Nations (ASEAN) way of handling matters (The Star, 1998):

> *We should uphold the Eastern culture where we do not make open criticisms which may lead to uneasiness and hurt the feelings of our neighbours. It will also create tension between us and this will be a bigger problem.*

Malaysians, as members of ASEAN, have been observed to be less forthright in expressing views and opinions and are uncomfortable in criticising and evaluating peers and subordinates (Asma Abdullah, 1996).

In this study, participants such as participants P5, P8, P9 who performed poorly in the TA sessions, reflected by the low ranking in TA, were positive in the interview and/or SUS, see Table 8.3. This inconsistency may be attributed to the fact that the participants did not want to comment negatively about the spreadsheet, as this would cause the experimenter to lose face. Furthermore, given that Malaysians try to refrain from giving negative comments, the participants gave positive comments instead to save the experimenter's face. This situation is similar to that described in Herman (1996) referred to in Section 7.2, about a participant's positive post-test interview response despite breaking down in tears while using the tool.

On the other hand, UAT responses of participants who were familiar with the experimenter were more consistent. Even if participants had problems in using the spreadsheet, (that is, low TA rank) the participants who were familiar with the experimenter commented negatively about *Hamparan*. This contrasting behaviour may be because they knew the experimenter and his disposition, and knew that negative comments would not cause him to lose face. For example, participant P17, who *was not* familiar with the experimenter interview response, was forthright but the participant mitigated his comments, probably to preserve face of the experimenter. " I think it's really difficult to learn. Ahhhh... no, actually its not... not that difficult...". On the other hand, participant P10, who *was* familiar with the experimenter, commented negatively (without mitigation) in the interview, "I wanted to enjoy it but I was like ... ahhh shocked... because there's, like, nothing there.. nothing... kind of *arahan* ("commands") until I really got *pening* ("a headache") ... I was taken aback... It wasn't that friendly I have to be frank with you. [Laughs]".

Another perspective on the results takes into account both the participants' computer experience and familiarity with the experimenter. The inconsistency in participants' responses may be attributed to both poor experience and unfamiliarity with the experimenter. Due to their unfamiliarity with Lotus 123, the participants experienced difficulty using the spreadsheet, and probably

harboured negative opinions of the spreadsheet. However, due to their unfamiliarity with the experimenter, and in order to preserve face, the participants provided positive rather than negative comments.

In contrast, participants with Lotus 123 (DOS) experience *and* familiar with the experimenter, were candid about the spreadsheet even though they had experienced difficulty in using *Hamparan*. Their forthrightness may be due to the fact that they were experienced computer users *and* that they knew the experimenter's feelings would not be hurt or that the experimenter would not lose face due the negative comments.

Explanations, reported in Section 8.5.1, relating to the consistency of data sets of participants experienced with Lotus (DOS) 123 were supported by the Kissel's (1995) study. For the collectivism explanation in Section 8.5.2, these explanations have been based on available literature pertaining to Malaysians' behaviour in collectivist countries. While the collectivism explanation has been reported in Yeo (2001), further support of the explanation was obtained from a Japanese usability expert, Professor Masaaki Kurosu. Professor Kurosu agreed with the explanations provided and added that he observed similar tendencies among Japanese subjects; Japan is also a more collectivistic than individualistic country (Kurosu, pers. comm., 2001). The author also asked whether Professor Kurosu agreed that the interview was the most inconsistent UAT. According to Professor Kurosu, the UATs are employed under a specific social situation which includes the relationship between the experimenter and subject, and is influenced by the testing situations, such as the testing location, or the physical environment. He said that of the three UATs, he believed the SUS might be the "most independent" from the usability evaluation situation, followed by TA, then the interview. The interview was dependent on the user evaluation situation as the participant was interviewed face-to-face, and thus those participants were under the highest "social pressure" (Kurosu, pers. comm., 2001). This social pressure refers to the culture which exists among Malaysians, such as the collectivistic behaviours mentioned above.

Even though the context concerned takes into account the fact that the usability evaluation was in Malaysia, the interaction between the experimenter and participants was an *intercultural interaction*. This intercultural interaction is

between the experimenter and the participant who are from two ethnic groups; the experimenter is a Chinese, and the participants were predominantly Malays; participants comprised fourteen Malays, two Indians and one Chinese. Thus, rather than values of a particular ethnic group, the values considered here are values common to all Malaysians (Asma Abdullah, 1996). Malaysia is a multi-ethnic society. The largest ethnic-groups are Malays (50% of the population), Chinese (30%) and Indians (10%). This distinction is made, as there is evidence that indicate different ethnic groups possess different work groups' values (Zabid, Anantharaman and Raveendran, 1997). However, the intercultural interactions in this study relate to the Malaysian culture in which the collectivistic behaviours prevail.

8.5.3 Power Distance

The high status participants made more negative comments as shown in the interview ranking in Table 8.4. It would appear that low participants appear to be less harsh in their comments unless they knew the experimenter. As mentioned in Section 7.2, a person of low status is unlikely to "go against" a person of higher status for fear of retribution. The exception may be that, the low status person may go against the person of higher status if the low status person knows the disposition of the high status person and that he or she would not "penalise" the lower status person for the negative feedback.

8.5.4 Collectivism and Power Distance

While data sets of high status participants' correlation coefficients were not as strong as the correlation coefficients of the participants "familiar with experimenter", there appears to be a relationship between the familiarity attribute (explained by collectivism) and status attribute (explained by power distance). According to Hofstede (1994), countries with large power distance are also likely to be countries, which are more collectivistic. This relationship, presented in Figure 8.2, is explained by the fact that people who are dependent on in-groups are also people who are usually dependent on power figures (Hofstede, 1994). For example, "most extended families have patriarchal structures with the head of the family exercising strong moral authority." (Hofstede, 1994).

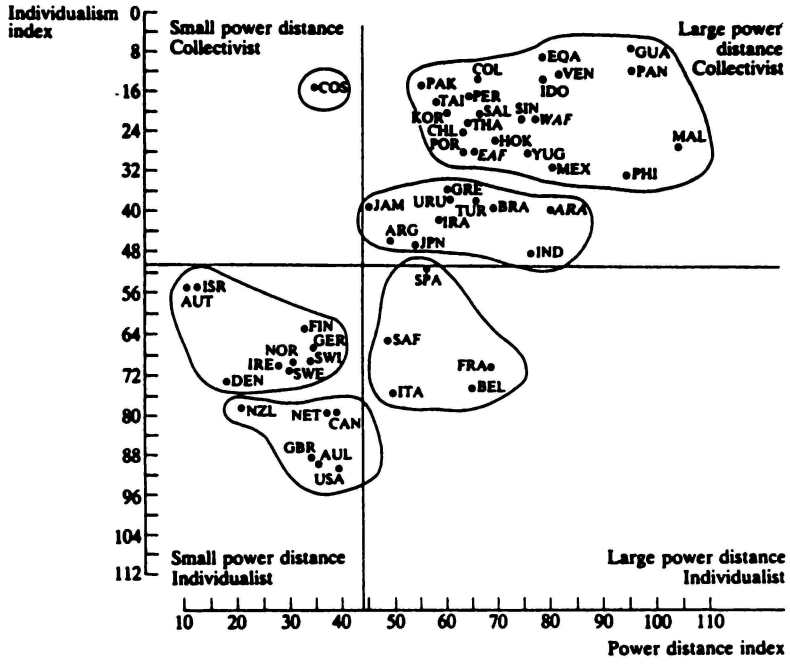

Figure 8.2: PD & IDV dimensions (sourced from Hofstede (1994))
PD: Power Distance, IDV: Individualism. For country name abbreviations see Table 8.6 below.

Abbreviation	*Country or Region*	*Abbreviation*	*Country or region*
ARA	Arab-speaking countries (Egypt, Iraq, Kuwait, Lebanon, Libya, Saudi Arabia, United Arab Emirates)	ISR	Israel
		ITA	Italy
		JAM	Jamaica
		JPN	Japan
		KOR	South Korea
ARG	Argentina	MAL	Malaysia
AUL	Australia	MEX	Mexico
AUT	Austria	NET	Netherlands
BEL	Belgium	NOR	Norway
BRA	Brazil	NZL	New Zealand
CAN	Canada	PAK	Pakistan
CHL	Chile	PAN	Panama
COL	Colombia	PER	Peru
COS	Costa Rica	PHI	Philippines
DEN	Denmark	POR	Portugal
EAF	East Africa (Ethiopia, Kenya, Tanzania, Zambia)	SAF	South Africa
		SAL	Salvador
EQA	Equador	SIN	Singapore
FIN	Finland	SPA	Spain
FRA	France	SWE	Sweden
GBR	Great Britain	SWI	Switzerland
GER	Germany F.R.	TAI	Taiwan
GRE	Greece	THA	Thailand
GUA	Guatemala	TUR	Turkey
HOK	Hong Kong	URU	Uruguay
IDO	Indonesia	USA	United States
IND	India	VEN	Venezuela
IRA	Iran	WAF	West Africa (Ghana, Nigeria, Sierra Leone)
IRE	Ireland (Republic of)	YUG	Yugoslavia

Table 8.6: Abbreviation of countries/regions (sourced from Hofstede (1994))

Schermerhorn and Bond (1997) argue that Malaysians exhibit combined behaviours of a collectivist *and* high power distance followers, rather than possessing only behaviours of people from collectivist countries, or possessing

only behaviours of people from a high power distance country. Schermerhorn and Bond (1997) suggests that the Malaysian culture is a reflection of influences of both cultural dimensions, which is displayed in behaviours such as public deference conformity, and politeness, no public expression of disagreement, as opposed to only collectivistic behaviours (team oriented and interactive) and only high power distance (respectful and responsive).

Thus, there may be interactions between the two cultural dimensions, collectivism and power distance. However, an analysis based on the combined attributes of familiarity and status was not carried out in this study. The sample size taking into account familiarity and status characteristics would be too small for correlation coefficients to be calculated.

8.6 Implications for Usability Evaluation in Malaysia

It would appear that the inconsistencies of the data sets may be due to the participants' characteristics. These characteristics include:

- experience of the participants (less experienced participants gave more positive but less accurate comments as they were not-knowledgeable about computers and software),
- preservation of face (given that Malaysia is a collectivistic society, maintenance of harmony is important, thus, participants gave positive comments), and
- respect for hierarchy (Malaysia is a high power distance country, with low status participants unwilling to give negative comments to high status members).

It is suggested that these findings need to be taken into account when conducting usability evaluation in Malaysia or, countries with similar cultural background.

From the findings, in order to ensure the tools employed in usability evaluation are effective, the imported UATs may have to be *localised* for the Malaysian context. In which case, the UATs, which collect objective measures, should be used as these techniques appear to be less influenced by factors such as preservation of face; participants do not have to go face-to-face with the experimenter. However, if only subjective UATs such as interviews are available, participants with these attributes should be recruited:

- people who are experienced in tools similar to the product being evaluated,
- participants who are familiar with the experimenter, or
- participants who are of higher status than experimenter.

If the product being evaluated possessed poor usability, the participants with above characteristics are more likely to provide accurate (forthright) feedback than other participants with other attributes. However, by selecting only participants with the above characteristics, it should be remembered that these participants would comprise only part of the target groups of users. As such, it is more appropriate to adapt the UATs such that accurate feedback can be obtained. (Most of this chapter has been reported in Yeo (2001)).

8.7 Conclusion of the Re-examination

The objective of this re-examination of data was to ascertain whether participants' characteristics may have contributed to the anomalies of the UAT data, that is, the inconsistency of the UAT data sets. It was speculated that the status of the participants may be one of the contributing factors.

It was found in the re-examination of the data that data sets from "participants with Lotus 123 (DOS) experience" and "participants who were familiar with the experimenter" agreed. On the other hand, the data sets of "participants without Lotus 123 (DOS) experience" and "participants who were not familiar with the experimenter" did not agree.

Thus, in relation to second part of the key question, the UATs are appropriate to ascertain the usability of the localised spreadsheet *provided* participants with the above mentioned characteristics are recruited for the usability evaluation. This *localisation* of the UAT would ensure that the data collected using the modified UATs would more reliably indicate the usability of the product being evaluated.

8.8 Summary

In this chapter, further analysis showed that the anomalous data could be attributed to the participants' cultural and technical background. Specifically, the

data collected from the three UATs were consistent if participants with the following attributes were used: participants who were familiar with the experimenter, participants who were expert in tools similar to the tools being evaluated. However, the data collected from the three UATs disagreed if participants with the following attributes were used: participants who were not familiar with the experimenter, and participants who were not expert in the tools similar to the tools being evaluated.

These results in particular have implications on how usability evaluation is conducted in Malaysia and countries which have similar cultural background to Malaysia. However, these results also have implications for the whole global-SDLC. These implications are described in Chapter 9.

Chapter 9 Global-SDLC Revisited

In the previous chapter, a re-examination of the usability data suggested that the anomalous data sets may be attributed to the participants' technical and cultural background. In particular, data collected using usability assessment tools (UATs) employing participants with certain characteristics may be more reliable than data collected employing participants without these characteristics. The implications of the findings on how usability evaluation is conducted in Malaysia and in countries whose people share similar cultures as Malaysians, were also described in the previous chapter. In this chapter, implications of the findings on the global-software development life-cycle (global-SLDC) and the claim made in Chapter 1 are considered.

9.1 Internationalisation and Localisation of Non-software

As presented in Chapter 8, the findings in this study provide evidence that suggests the imported UATs can be employed in the formative usability evaluation of the localised software. The findings also suggest that imported UATs can be employed in the summative evaluation of the localised software, *provided* the usability assessment process is localised to match the cultural characteristics of the target users of the software being evaluated. The usability assessment process (UAP) consists of the tools, such as the UATs, protocol, location in which the usability evaluation is conducted, people (includes experimenters, and participants), as well as these people's culture. With the localisation of the UAP, there is a likelihood of obtaining more reliable results than in a usability evaluation without the adaptation.

The following example is provided to illustrate this adaptation of the UAP. Imagine the scenario: a localised software application is to be evaluated in Thailand, which has similar cultural background as Malaysia. The target users of the localised application are data-entry operators. These operators are considered users of low status. (Low status people refers to those without a Bachelor or higher degree.) The skill levels of these operators are low. They are trained only for specific tasks using specific software. In usability evaluation, the experimenter would like to use interviews to collect as much information about

the participants' experience using the new software. In this case, a possible localisation strategy would be for the experimenter to establish a good rapport with the participants, so that the participants are familiar with the experimenter. As reported in the last chapter, usability data collected from participants who were familiar with the experimenter were more reliable than data collected from participants who were not familiar with the experimenter. Furthermore, to counter potential problems with status differences, the experimenter could be introduced as someone of the same status as the data-entry operators. Thus, the localisation of the UAP to match the participants' attributes could improve the chances of obtaining more reliable data.

As described in the above example, the UAP could be localised. In fact, the usability evaluation could be first internationalised and then localised. Suppose that usability data of a software application has to be collected from many cultural groups. In the internationalisation process, the culture-dependent components of the usability evaluation are isolated. An instance of a culture-independent component is the usability evaluation's goal, that is, to collect data on the usability of the software being evaluated. This goal would be applicable to all target cultural groups. An instance of culture-dependent component is the UAP. This UAP would have to be localised for each of the specific cultural groups. For instance, the UAT employed, as part of the usability assessment process (UAP), could be the interview. When the interview is employed in a particular cultural group's context, the UAP (a culture-dependent component) is localised to fit that target cultural group's context. The interview questions as part of the UAP may be localised/translated to the target cultural group's language. In a different cultural group, interviews may not collect data in a useful form. Thus, the UAP would have to be localised for the new group; another UAT such as focus groups, may be used instead of interview. The goal (culture-independent component) of usability evaluation would remain the same, that is, to collect usability data of the software.

The above example illustrates how the usability evaluation phase can be internationalised and localised. The example also reveals that the internationalisation and localisation process can also extend beyond software. Currently, internationalisation and localisation involves only software. Based on the findings of this study, and the example presented above, the usability

evaluation of the global-SDLC can also be internationalised and localised. Figure 9.1 depicts the modified global-SDLC, in which internationalisation and localisation of the usability evaluation phase are included in the global-SDLC.

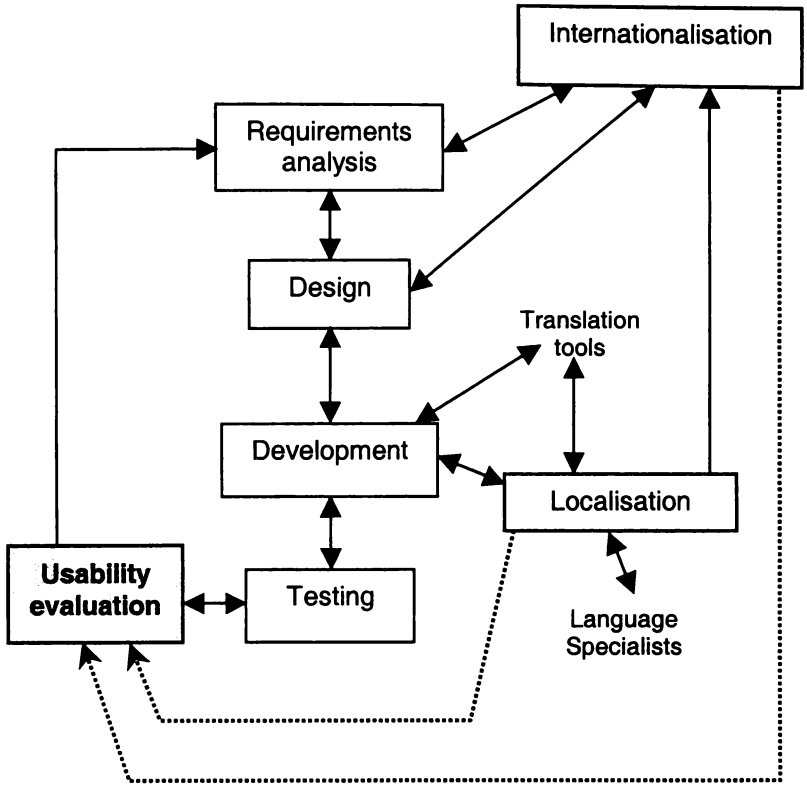

Figure 9.1: Modified Global-Software Development Life-cycle
The dashed lines indicate that the internationalisation and localisation process may also be required for the usability evaluation phase.

9.2 Importance of Effective Usability Evaluation in Global-SDLC

In this study, imported UATs were employed in the usability evaluation of an internationalised and localised tool. The results of the research indicate certain UATs, when employing participants with specific attributes, may provide inconsistent results. This result further emphasises the importance of effective usability evaluation in global-SDLC. The significance of the effective usability evaluation is described next.

Software targeted at the different cultural groups could be developed by following the global-SDLC guidelines. Once the software has been internationalised and localised, the success of the localised software is not known until representative users assess the software application in the usability

evaluation phase. If the UAP employed in the usability evaluation phase is effective, the usability of the software can be determined. However, if the process employed in the usability evaluation phase is ineffective, the success of the localised software is not known. For example, inaccurate data from ineffective usability assessment process (UAP) may indicate the software to possess much better usability than is actually the case. Such an anomaly was illustrated by the participants' interview comments in this study, which were incongruent to the other usability data collected.

Thus, only with an effective usability evaluation phase can the success of the software/acceptability of the software be known. The success of the software is also dependent on the success of the processes (in the global-SDLC) applied to develop the software. Thus, effective usability evaluation is required to determine the success of all processes employed to develop the software. As such, the success or failure of the internationalisation and localisation process can only be determined when the usability tools employed in the usability evaluation phase are reliable and effective. Software engineers require an effective usability evaluation phase to provide data which not only inform them of the success and acceptability of the software developed by the users, but also the success or failure of the global-SDLC process employed to develop the software.

To sum up, effective usability evaluation is important to the global-SDLC. Only with effective usability evaluation would software engineers be able to determine the success of the software and ascertain if the global-SDLC process is effective.

9.3 Global-SDLC has to be Adapted

Based on the findings of this research, there is evidence to suggest that the usability assessment process (UAP) employed in the usability evaluation phase may be influenced by the participants' culture. As described in Section 9.1, the UAP may have to be adapted to fit the target groups' cultural characteristics. The other phases of the global-SDLC may also be influenced by cultural factors. Like the UAP, the global-SDLC is an invention of the developed nations, and thus may have to be adapted to fit the target cultural groups when applied outside the developed nations. It is suggested that other phases of the global-SDLC may also

be affected by the cultural context in which the global-SDLC process is employed. The cultural context includes all the stakeholders (people involved in the software development, such as clients and developers), culture of the stakeholders, tools, techniques, environment, and organisations involved in the global-SDLC process. This suggestion is supported by Thanasankit and Corbitt (2000) who state that the Thai culture impacts on the requirements gathering process in Thailand.

Thanasankit and Corbitt (2000) note that because Thailand is a high power distance country, the hierarchical nature of the Thai society and especially the construction of responsibility at the top contributed to delays and inhibitions in getting requirements approval. Each time the Thai systems analysts wanted to proceed to collect more requirements to design a system, they needed to get approval from their clients to go ahead. However, the client's approval normally has to come from committees in the client organisation. These committees usually checked the requirements and then set up a formal meeting to confirm them. Further delays may result when the committee hung on the requirements, which was often, and then passed the requirements on to a Steering Committee for approval or to add more requirements. Thanasankit and Corbitt (2000) believe that understanding of the Thai culture would influence successful design, development and acceptance of new information systems in Thailand.

9.4 The Claim Revisited

The claim made in Section 1.5 was that the global-software development process, as currently defined, has to be adapted in its entirety for the target cultural group before it can be employed in migrating software from a source cultural group.

The global-software development process is the process which employs the global-SDLC to produce software for multiple cultural groups. This global-SDLC is the conventional life-cycle which incorporates the internationalisation and localisation processes.

As noted in Section 9.1 of this thesis, it was found that the usability assessment process (UAP) of the usability evaluation phase of the global-SDLC may have to be adapted to the target cultural group. Given that the usability evaluation process has to be adapted to specific cultural groups, it would appear

that any process with human interactions would require some form of adaptation. Usability evaluation, itself a process, is also a sub-process of the global-software development process.

Extending this, if adaptation of the sub-process is required, then all sub-processes with human interactions in the software production will need adaptation – such as the requirements analysis (sub-process) reported in Thanasankit and Corbitt (2000). By induction, since all sub-processes involving human interactions require some form of adaptation, the entire global-software development process (which is an assembly of all the sub-processes) also would need to be adapted to the target cultural context.

However, before attempting to adapt the whole global-software development process, the whole process should be evaluated to establish its effectiveness in the target cultural context. This examination is essential, because it may not be necessary to adapt all phases. As found in this study, some of the imported UATs seem to be appropriate for part of the usability evaluation process. Thus, an evaluation of the global-SDLC's (employed in the global-software development process) effectiveness would uncover further areas that may require adaptation.

The claim made in this thesis is thus partially supported. The whole global-software development process may not have to be adapted. However, what is important is the evaluation of the imported tools. To paraphrase Nielsen (1990b), the imported global-SDLC process when used in another cultural context other than the one where it was developed and for which it was designed, is a "new" global-SDLC. The "new" global-SDLC must be evaluated to establish its effectiveness in the new cultural context. Furthermore, target users must be involved in the evaluation process. Further research is required to examine all phases and investigate the possible effects of cultural factors on the operation of the phases in the development of software for multiple cultural groups.

A possible extension to adapting the global-SDLC to target cultural groups is the internationalisation and localisation of global-SDLC tools and processes. The advantage of internationalising the global-SDLC is that, each time the global-SDLC is applied to a new context, the whole global-SDLC need not be adapted. Only the culture-dependent component needs localisation. Further work

is required to determine whether there are any benefits to the internationalisation and localisation of global-SDLC.

9.5 Summary

This chapter has shown the implications of the results of this research on the global-SDLC as described in current literature and has highlighted the importance of usability evaluation in the global-SDLC. The suggestion is made that internationalisation and localisation be extended to the usability evaluation phase of the life-cycle, such as the usability assessment process. Furthermore, the whole global-SDLC *may* have to be adapted to suit the target cultural group. The global-SDLC should, however, be evaluated first in the target cultural context. The next chapter presents the conclusions and suggestions for further work arising out of this thesis.

Chapter 10 Conclusions

The previous chapter described the implications of the study's result for the usability evaluation phase and for the global-SDLC. In this chapter, general conclusions drawn from the findings are reported in Section 10.1. Section 10.2 provides a summary of the contributions of this research and Section 10.3 describes further work which could be carried out to extend this research.

10.1 General Conclusion

Based on the results of this research, tools brought in from foreign countries which are employed to assess the usability of software are not directly suitable for use in the importing nation.

The global-SDLC employed to develop software for multiple cultural groups may also for similar reasons not be directly applicable to the importing nation.

In order to be useful, these tools and the software development process must first be evaluated to establish the tool's effectiveness in the target cultural context. The tools may themselves have to be adapted to the target cultural group. Only after adaptation of the imported technologies to suit the target cultural group, can these technologies be used with confidence in the technologies' effectiveness in the target nation. Otherwise, as noted in Day's (1999) treatise, the tool may be more of an impediment than an aid in advancing the goal of software development of software to multiple cultural groups.

This finding is of particular importance to developing nations, such as Malaysia, which import technology. Ideally, developing nations should be creating their own software. But until these nations possess an established software industry which can meet their software demands, they will have to import computer-based technologies. However, based on the results of this study it is suggested that all imported technologies should be first examined and appropriately modified for cultural adaptation needs before applying the technologies.

10.2 Contributions

This section provides a summary of conceptual and practical contributions made by this exploratory study.

10.2.1 Conceptual Contributions

This research extends knowledge of usability evaluation of software migrated from a source to a target cultural group.

The study provides a framework for the target population of developing nations to examine whether imported tools, employed in the usability evaluation, can be used for the migration of software to target cultural groups. This study adds to the body of knowledge on the development of software for developing countries.

An imported usability assessment process can be employed in the usability evaluation of a software application (adapted from a source cultural group to target cultural group), *provided* the usability assessment process is localised to suit the target group's cultural attributes (Section 9.1). The research finding suggests that imported usability assessment process comprises not only the UATs, the protocol, the experimenter, and the participants, but also the cultural behaviour of the groups to which the experimenter and participants belong (Section 9.1).

The research suggests that the internationalisation and localisation processes may extend beyond software, that is, on to the usability assessment process and possibly the global-SDLC (Section 9.1).

The research also suggests that the global-SDLC process may have to be adapted for each target cultural group (Section 9.3). The research also highlights the importance of evaluation on not just the software, but the very processes employed in developing the software (Section 9.2 and Section 9.3).

The research highlights a need to evaluate *all* imported tools and phases of the global-SDLC which involves human interaction to ensure that the tools and processes are effective in the importing nation (Section 9.4 and Section 10.1).

10.2.2 Practical Contributions

The results of this study are applicable to Malaysia and developing nations which share similar cultural backgrounds. As such, think aloud and interview may be effective in identifying usability problems in spreadsheets. The think aloud, SUS, and interview were also found to be appropriate in the usability evaluation, only if suitable target users and experimenter are employed. Also, the experimenter would need to be aware of possible anomalies in the interview data.

This thesis provides a description of the whole global-SDLC, which was suggested as being incomplete in Section 2.6.6. The description was aggregated from a variety of sources (Section 3.3). The thesis also reports on an application of the global-SDLC in the adaptation of a simple spreadsheet (Section 4.1). As a result, an experimental spreadsheet with a Bahasa Melayu and Iban interface was developed.

10.3 Further Work

This section suggests aspects of the work reported which might be useful for further research.

10.3.1 More Cost-Effective UATs Required

In research, despite the reported and observed high costs of the analysis of the think aloud data (see Section 7.1.2), the employment of the think aloud method to collect qualitative data is justified. However, in the "real world", more cost-effective methods are required. For example, a possible research direction could be to examine a modified System Usability Scale questionnaire. The SUS used in this study, consists of only closed-ended questions, which were not useful in collecting the qualitative data required for the improvement of the spreadsheet. A questionnaire with open-ended questions might be a more appropriate tool to elicit the required type of data. Further studies, in Malaysia, or elsewhere would be required to determine whether such a questionnaire would be a more efficient tool in uncovering usability problems.

10.3.2 Results Applicable in Malaysia Only

It must be emphasised that this study was conducted in Malaysia and that the results may only be valid in Malaysia and possibly in countries which share similar cultural characteristics as Malaysia. The results of this study are not

necessarily applicable to all developing nations. To explore whether the UATs are effective in the other developing nations, the imported usability assessment processes must be evaluated in the target cultural group's environment. Furthermore, representative users must be employed in the evaluation process. A larger sample size of representative users should also be employed in the study in order to obtain results which are more reliable and valid.

10.3.3 Confirmation of Cultural Effects

One possible explanation, provided in Section 8.5, to account for the anomalies of the data sets was based on the participants' cultural characteristics. An important research direction would be to ascertain whether the participants' characteristics were the actual cause of the anomalies.

As the participants' characteristics were associated with the collectivism and power distance cultural dimensions (see Section 8.5), this study should be replicated in other collectivistic and high power distance countries such as Thailand, to confirm the results of this study. This study should also be carried out in a country such as Sweden, which is an individualistic and low power distance country. Compared to Malaysia, Sweden appears in the opposite end of the two cultural dimensions. The comparison of the results of Malaysian, Thai, and Swedish study would be able to shed more light on the cause or causes of the anomalous data.

10.3.4 Localisation of Usability Assessment Process

As suggested in Section 9.1, the usability assessment process may have to be adapted to suit a particular target cultural group. Further investigation is required to identify ways of localising the usability assessment process. This investigation would include identifying the sets or combination of components of usability assessment process that work. By conducting such studies, a larger set of effective usability assessment processes would be created and could be made available to developers. The appropriate usability assessment process can then be selected from that set of potential usability assessment processes depending on the context of the usability evaluation.

10.3.5 Validating Each Phase in the Global-SDLC Process

Given that the results of this research indicate that the usability evaluation phase may be influenced by cultural factors, other phases of the global-SDLC may

also be affected. There would seem to be a need to research the role that each of the phases of global-SDLC when employed in providing software for different cultural groups. In particular, special attention should be given to the study of the requirements analysis and design phases. These phases are suggested because errors in these phases may lead to high costs in fixing software (Mynatt, 1990).

10.3.6 Validating the Whole Global-SDLC as a Culture-specific Process

In addition to validating the processes as described in the previous section, the whole global-SDLC may need to be examined to determine its effectiveness in the development of software for specific cultural groups.

This examination is suggested as being necessary because if any of the phases in the global-SDLC are affected by cultural factors, the cultural effect may also impact not only that phase but also the whole global-SDLC. For example, suppose it was found that a story-telling method was needed to collect data in the requirements analysis phase in Mongolia. The type of data collected may impact on the other phases in the global-SDLC. Anthropologists may be required to interpret the data collected before software designers can use the information collected.

10.4 Summary

This study showed that the usability evaluation employed as part of the global-SDLC could be a sound process for providing software from a source to target cultural groups, provided the usability assessment process is localised to suit the target users' attributes. When sets of suitably localised usability assessment processes are developed, the global-SDLC could also provide a foundation for creating software suitable for multicultural societies. The localisation, however, includes not just the usability assessment process, but may extend to the whole global-SDLC.

A general conclusion arising from the research is that technologies cannot be unreflectively applied to target cultural groups. The technologies must be evaluated for their effectiveness and modified appropriately for each target cultural group if the members of the group are to find the technologies acceptable and usable.

Appendices

Appendix A User Guide for TCALC

This user guide provides information about the commands on using TCALC. This User Guide assumes that users have had some experience using a computer.

Conventions used

- Letters that are bolded are keys to be pressed, for example, **/-L-S** means press/, followed by L and S.
- <Return> refers to the Carriage Return key
- <CTRL>-**L** refers to pressing the Control key (and without letting go) pressing **L**, i.e. the two keys are pressed simultaneously.
- <ESC> refers to the escape key

Getting Started

- Type **TCALC** at the DOS prompt, and press <Return>

- TAKE NOTE
  - ⇒ Press / to obtain the list of commands in TCALC.
  - ⇒ <ESC> will cancel most commands, except when you are saving the spreadsheet–refer to the Quit section in the Command list below.
  - ⇒ Press **Y** for Yes and **N** for No.
  - ⇒ Formula **A1+A2+A3** is equivalent to **A1:A3.** There is no sum() function.

- WARNING

  This spreadsheet has no UNDO. Ensure all the changes you make to your files in TCALC are saved into a file.

Hierarchy of Spreadsheet Menus

Spreadsheet	Format	Delete	Goto	Col	Row	Edit	Utility	Auto	Quit
Load				Insert	Insert		Recalc		
Save				Delete	Delete		Formula display		
Print				Width					
Clear									

Commands Summary

The list below is a summary of the available commands in TCALC. The commands are listed according to the Menus as depicted in the Hierarchy of Spreadsheet menus above.

SPREADSHEET

Load: Loads a spreadsheet you specify.

Save: Saves the current spreadsheet you are working on.

Print: Prints the spreadsheet to a file or directly to a printer.

Clear: Clears the contents of the whole spreadsheet.

FORMAT
Allows you to change the justification, the dollar format and number format.

DELETE
Deletes the cell contents.

GOTO
Go to another cell.

COL
Insert: Inserts a column. All columns to the left of cursor are shifted one column to the left.

Delete: Deletes a column. Deleted the column the cursor is on.

Width: Changes the column width.

ROW
Insert: Inserts a row. All cells row on and below the cursor are shifted down.

Delete: Deletes a row. Deletes the row the cursor is on.

EDIT
Allows the user to edit the cell content. Short cut key is F2.

UTILITY
Recalc: Allows the user to carry out a recalculation when automatic recalculation is turned off.

Formula display: Allows the user to toggle formula display.

AUTO
Turns on or off the automatic recalculation.

QUIT
Quits from the spreadsheet. **WARNING**: Pressing <ESC> after you have just pressed Quit, will not stop the Quit action. The spreadsheet file will not be saved.

How to use the commands

Pressing <ESC> will stop the commands you have given.

The following paragraphs are organised according to the order of the Commands that appear when you press **/**. After you have pressed /, and you press S for spreadsheet, the Load Save Print and Clear commands will appear.

SPREADSHEET

Load: Loads a spreadsheet you specify.

- On pressing **L**, you see the following message:
  Enter the file name of the spreadsheet:
- Enter the name of the spreadsheet you wish to load, and the path if required.
  Note: DOS filenames are only eight characters long with a 3 character-extension.

Save: Save the spreadsheet

- On pressing **S**, you will see the following message:

  Enter the file name of the spreadsheet:
- Enter the name of the spreadsheet you wish to save as, and press <ENTER>. Note: DOS filenames have a maximum of eight characters with a 3 character-extension.
- If a file of the same name exists, the following message will appear:

  The file exists. Do you want to overwrite it?
- Type **Y** is you wish to overwrite the file, and **N** if you don't. If you typed N, you will need to carry out the saving procedure again, that is / for the commands, S for spreadsheet, S again for Save.

Print: Prints the spreadsheet to a file or directly to a printer

- On pressing **P**, you will be see the following message:

  Enter the file name to print to, or press ENTER to print on the printer.
- If you wish to print, to a file, type in the filename. The spreadsheet will be saved as a text file. Note: DOS filenames have a maximum of eight characters with a 3 character-extension.
- Press <ENTER> if you wish to print to the printer.

Clear: Clears the contents of the whole spreadsheet

- On pressing C, the spreadsheet will clear all the contents from the spreadsheet.

FORMAT

Allows you to change the justification, the dollar format and number (adding commas) format.

- On pressing F, you will be prompted with the following message:

  Enter the first cell to format:
- You will need to type in the first cell address (of a range) and press <ENTER>. The following message will appear:

  Enter the last cell to format:
- Type in the last cell address of the range and press <ENTER>. The following message will appear:

  Do you want the cell right-justified?
- Type **Y** if you wish the cell to be right justifies, **N** if you do not.

  Do you want numbers in a dollar format?
- Type **Y** if you want the numbers with the currency format ($). You will be prompted:

  Do you want commas in numbers?
- Again type **Y** if you do, and **N** if you do not wish to have commas.

DELETE

Deletes the cell contents

- On pressing D, the contents of the cell you are currently in will be deleted.

GOTO

Go to another cell

- On pressing G, you will see the following message:
  Enter the cell to go to:
- Enter the cell address you wish to go to.

COL

Insert: Inserts a column

- On pressing **I**, a column will be inserted where the cursor was located. All the other columns will be shifted to the right.

Delete: Deletes a column

- On pressing D, the column on which the cursor was on will be deleted,

Width: Changes the column width

- On pressing W, you will be prompted with the following message:
  Enter the new column width:
- Type in the new column width. The default column width is 8.

ROW

Insert: Inserts a row.

- On pressing **I**, all cells rows on and below the cursor are shifted down.

Delete: Deletes a column. Deletes the row the cursor is on.

- On pressing **D**, the column on which the cursor was on will be deleted.

EDIT

Allows the user to edit the cell content. Short cut key is F2.

- On pressing **E**, you should be able to change the contents. Use the Right or Left arrow key to move the cursor.
- When you have finish editing, press <ENTER>.

UTILITY

Recalc: Allow the users to carry out a recalculation. When AutoCalc is off.

- On pressing **R**, the cell value where formulae reside will be updated or recalculated. This will work when AutoCalc is off. If AutoCalc is on, the recalculation of any formulae is carried out automatically as soon as any changes are made.

Formula display: Allows the user to turn on and turn off the formula display.

- On pressing **F**, all formulae in the spreadsheet will be shown instead of the values. "Formula" will appear at the top (in purple) when this option is on.

AUTO

Turns on or off the automatic recalculation.

- On pressing **A**, automatic recalculation will be turned off. "AutoCalc" will appear at the top right of the spreadsheet if automatic recalculation is turned on.

QUIT

Allows users to save the file. WARNING: Pressing <ESC> (after you have just pressed Quit) will not stop the Quit action. The spreadsheet file will not be saved.

- On pressing **Q**, you will be prompted with the following message if you had made some changes which were not saved:

  Save current spreadsheet?
- Type **Y** and <ENTER> if you wish to save the spreadsheet. You will then be prompted:

  Enter the file name of the spreadsheet:
- Enter the name of the spreadsheet you wish to save as and press <ENTER>.
- If a file of the same name exists, the following message will appear:

  The file exists. Do you want to overwrite it?
- Type **Y** is you wish to overwrite the file, and **N** if you don't. DO NOT press <ESC> as this will stop the FIRST program.
- Type **N** if you do not wish to save. TCALC will then quit.

Appendix B Message Tokens

This appendix contains the message tokens for English, Bahasa Melayu, Māori and Iban.

B.1 Message Tokens (English)

```
1     "English: Spreadsheet"
2     "Press E to select English"
3     "Can't open the file"
4     "Press / for the list of commands"
5     "Memory Available:"
6     "ERROR"
7     "Not enough memory to allocate cell"
8     "Empty"
9     "Text"
10    "Value"
11    "Formula"
12    "AutoCalc"
13    "Formula"
14    "Enter the file name of the spreadsheet:"
15    "Press any key to continue"
16    "Enter the new column width:"
17    "The file exists.  Do you want to overwrite it?"·
18    "Not enough memory for entire spreadsheet"
19    "That is not a FIRST spreadsheet"
20    "The file does not exist"
21    "Enter the cell to go to:"
22    "You must enter a number from %d to %d."
23    "That is not a legal cell"
24    "Enter the first cell to format:"
25    "Enter the last cell to format:"
26    "The row or the column must be the same"
27    "Do you want the cell right-justified?"
28    "Do you want numbers in a dollar format?"
29    "Do you want commas in numbers?"
30    "How many decimal places should the number be rounded to?"
31    "Do you want to print in 132 columns?"
32    "Enter the file name to print to, or press ENTER to print on
      the printer"
33    "Print the border?"
34    "Loading..."
35    "Saving..."
36    "Save current spreadsheet?"
37    "Parser stack overflow."
38    "Spreadsheet, Format, Delete, Goto, Col, Row, Edit, Utility,
      Auto, Quit"
39    "SFDGCREUAQ"
40    "Load, Save, Print, Clear"
41    "LSPC"
42    "Insert, Delete, Width"
43    "IDW"
44    "Insert, Delete"
45    "ID"
46    "Recalc, Formula display"
47    "RF"
48    "YN"
49    "$"
```

B.2 Message Token (Bahasa Melayu)

```
1     "Bahasa Melayu: Hamparan"
2     "Tekan B untuk Bahasa Melayu"
3     "Fail tidak dapat dibuka"
4     "Tekan / untuk senarai perintah"
5     "Memori yang sedia ada"
6     "Ralat"
7     "Tidak cukup memori untuk memperuntukkan sel"
8     "Kosong"
9     "Teks"
10    "Nilai"
11    "Rumus"
12    "PerhitunganAuto"
13    "Rumus"
14    "Taipkan nama fail untuk dimuatkan"
15    "Tekan mana-mana kekunci untuk menjalankan hamparan"
16    "Taipkan lebar lajur baru:"
17    "Fail tersebut wujud.  Mahukah anda menulisgantikan fail
      lama?"
18    "Memori tidak mencukupi untuk seluruh hamparan"
19    "Itu bukan fail hamparan FIRST"
20    "Fail tersebut tidak wujud"
21    "Masukkan sel yang diingini:"
22    "Anda mesti masukkan nombor dari %d hingga %d"
23    "Sel tersebut tidak sah"
24    "Masukkan sel pertama untuk diformat:"
25    "Masukkan sel terakhir untuk diformat:"
26    "Baris atau lajur mestilah yang sama"
27    "Adakah anda inginkan semua sel diselaraskan ke kanan?"
28    "Adakah anda ingin semua nombor dalam format RM?"
29    "Adakah anda ingin semua nombor berkomma?"
30    "Berapa tempat perpuluhan yang perlu dibundarkan untuk
      setiap nombor?"
31    "Mahukah anda mencetak 132 lajur?"
32    "Masukkan nama fail untuk dicetak, atau tekankan ENTER untuk
      mencetak."
33    "Cetakkan border?"
34    "Dalam proses memuatkan fail ..."
35    "Dalam proses menyimpan ..."
36    "Simpan hamparan ini?"
37    "Tindanan penghurai melimpah atas"
38    "hamparaN, Format, Hapus, Goto, Lajur, Baris, Sunting,
      Utiliti, Auto, Keluar"
39    "NFHGLBSUAK"
40    "Muatkan, Simpankan, Cetakkan, Hapuskan"
41    "MSCH"
42    "Sisipkan, Hapuskan, Lebar"
43    "SHL"
44    "Sisipkan, Hapuskan"
45    "SH"
46    "Hitung semula, Tunjukkan rumus"
47    "HT"
48    "YT"
49    "RM"
```

B.3 Message Tokens (Māori)

```
1     "Māori: Te Ripanga"
2     "Patotōhia M kia Māori"
3     "Kāore e taea te whakatūwhera i taua kōnae."
4     "Patotōhia / kia kitea ngā tono"
5     "Mahara wātea:"
6     "KUA HEE"
7     "Kua pau kē te wāhi mahara"
8     "Kore kau"
9     "He Kōrero"
10    "He Nama"
11    "He Tātai"
12    "KaMahi Tātai"
13    "Whakatakotoranga"
14    "Tuhia te ingoa o te ripanga"
15    "Patotōhia tētehi pātuhi kia haere tonu"
16    "Tuhia te whānui hou o te kapa"
17    "Kua whakatŪngia kētia taua kōnae.  Me whakakore i te kōnae
      tawhito?"
18    "E kore e taea te katoa o tēnei ripanga, ka pau kē te wāhi
      mahara"
19    "Ehara tēnei i te kōnae Ripanga FIRST"
20    "Kāore kau he kōnae e mau ana i taua ingoa"
21    "Ka haere ki tēwhea pŪtau?"
22    "Tuhia tētehi nama i waenganui i te %d me te %d"
23    "Kāore e pai ana tēnei nama pūtau"
24    "Tuhia te pūtau tuatahi hei whakarite"
25    "Tuhia te pūtau whakamutunga hei whakarite"
26    "Kia ōrite ai te rārangi, te kapa rānei"
27    "Kia whakataka te pūtau ki te taha katau?"
28    "Kia whakarite moni ngā nama?"
29    "He ..kei waenga i ngā nama?"
30    "Kia hia ngā mati whaïra?"
31    "Kia 132 ngā kapa i te mahi tā?"
32    "Ka tukua ki tēwhea kōnae (patotōhia te ENTER anake hei tā)"
33    "Me tā hoki i ngā pae pūtau?"
34    "Taihoa, kei te whakatitea..."
35    "Taihoa, kei te tiaki..."
36    "Me tiaki i tēnei ripanga?"
37    "Kua pokea e te rāhi o ngā tono."
38    "riPanga,Wh'rite,wh'Ngaro,Haere, Kapa,Rārangi,wh'Tika,Mahi
      tātai,Ahua,wh'Oti"
39    "PWNHKRTMAO"
40    "Huaki, Tiaki, Mahi Tā, Whakangaro"
41    "HTMW"
42    "whakaUru, whakaKore, Whānui"
43    "UKW"
44    "whakaUru, whakaKore"
45    "UK"
46    "whakātu Tātai, whakātu Nama"
47    "TN"
48    "AK"
49    "$"
```

B.4 Message Tokens (Iban)

```
1     "Iban: Pengancau"
2     "Tekan ka I enti ka jaku Iban""
3     "Fail endah ulih dibuka"
4     "Tekan ka / enti ka isi kandung ngarah"
5     "Memori ki udah bisi"
6     "Penyalah"
7     "Enda cukup memori ke ngalai ke sel"
8     "Nadai utai"
9     "Tulis"
10    "Ungkus ki bërega"
11    "Rumus"
12    "NgitungKediri"
13    "Rumus"
14    "Pasuk ke nama fail"
15    "Tekan aja enti ka nyambung ngancau"
16    "Pasuk ka pemesai lujur baru"
17    "Fail to bisi. Ka nuan nganti fail ka lama?"
18    "Enda cukup memori kena pengancau"
19    "Nya ukai fail pengancau FIRST"
20    "Nadai fail ki bakai tu"
21    "Tama sel kä dë kä :"
22    "Nuan patut nama ka numbur ari %d ka %d"
23    "Enda betul sel nya"
24    "Tama ka sel ki terubah enti ka format:"
25    "Tama ka sel ki penghabis enti ka format:"
26    "Baris atu lujur mesti ka sama"
27    "Ka nuan ngasoh sel nya lurus mayang ba sepia kanan?"
28    "Ka nuan ngaso numbur nya dalam format RM?"
29    "Ka nuan ngasoh numbur nya bisi koma?"
30    "Berapa alai perpuluhan ti perlu dibundarkan ka tiap
      numbur?"
31    "Ka nuan cetak 132 lujur?"
32    "Tama ka nama fail ka dicetak alam atau tekan ka ENTER enti
      ka cetak?"
33    "Cetak ka sempadan?"
34    "Alam proses muat ka fail ..."
35    "Alam proses nyimpan ka fail..."
36    "Ka nuan nyimpan pengancau tu?"
37    "Tindanan penghurai melimpah atas"
38    "penGancau,Format,Ngelenyau,Kin
      ka,Lujur,Baris,Sunting,Utiliti,keDiri,Pansut"
39    "GFNKLBSUDP"
40    "Padat ka, nYimpan, Cetak ka, Ngelenyau ka"
41    "PYCN"
42    "Tama ka, Ngelenyau ka, peMesai"
43    "TNM"
44    "Tama ka, Ngelenyau ka"
45    "TN"
46    "Itung baru, Padah ka rumus"
47    "IP"
48    "AN"
49    "RM"
```

Appendix C Tools Used in the Experiments

C.1 Consent to Participate Form

This form was sourced from Shneiderman (1992).

Informed Consent

1. I have freely volunteered to participate in this experiment.
2. I have been informed in advance what my task(s) will be and what procedures will be followed.
3. I have been given the opportunity to ask questions, and have had my questions answered to my satisfaction.
4. I am aware that I have the right to withdraw consent and to discontinue participation at any time, without prejudice to my future treatment.
5. My signature below may be taken as affirmation of all the above statements, it was given prior to my participation in this study.

Name: .....

Signature: .....

Date: .....

C.2 Instructions to Conduct the Experiment

Instructions for the Think Aloud Experiments

The objective of this experiment is to see how users use this spreadsheet. Before you begin to use the software, I will get some information about your computer background. Then, you will complete the tasks listed here. After completing the tasks, I would like you to complete a questionnaire and ask you some questions about the spreadsheet.	Objektif eksperimen ini ialah melihat bagaimana hamparan ini digunakan. Sebelum itu, saya perlu mendapatkan latar belakang pengunaan komputer anda. Kemudian, anda akan menyempurnakan "task" yang disenaraikan. Anda juga dikehendaki mengisi satu "daftar pertanyaan" Selepas itu, saya akan tanyakan beberapa soalan mengenai perisian tersebut.
I will also need you to sign a consent form. This is to satisfy the requirements required by the University of Waikato.	Anda juga perlu menandatangani borang keizinan. Keizinan ini diperlukan oleh Universiti of Waikato.
Just remember that I will not be judging you but observing the way you interact with the software.	Ingatlah bahawa saya tidak menilai "performance" anda tetapi hanya melihat bagaimana anda menggunakan hamparan tersebut.

Questions to Ask/Prompt Subjects

Tell me what you are thinking? What are you thinking now?	Apakah yang anda sedang fikirkan?
What are you intending to do?	Apakah tujuan anda sekarang?
What do you hope to achieve with that move?	Apakah yang anda harap boleh dicapaikan dengan gerakan tersebut?
How do you think you can solve that problem?	Terangkan bagaimana anda boleh selesaikan soalan itu?
Can you guess what it means?	Boleh anda tekakan/ agakkan makna arahan tersebut?
What do you think that message means?	Pada pendapat anda, apakah makna ayat/arahan/message ini?
What do you think that message wants you to input?	Pada pendapat anda, apa yang perlu ditaipkan/input?
How do you know what to do next?	
Why did you choose to do that this way rather than some other way?	
What was your expected results?	

Interview Questions

What is your opinion of the spreadsheet you have just used?	Apakah pendapat anda mengenai hamparan ini?
Can you suggest any improvements to the software?	Bolehkah anda beri cadangan untuk memperbaiki/membetulkan hamparan?

C.3 Logged Data

The raw data recorded into a text file, keyst.log, is shown below.

```
0 0 0 98 63 5770 0 0 0 1234 66 5771 0 0 0 13 66 5943 0 0 0 336
66 6022 0 0 1 82 66 6060 0 0 1 101 66 6064 0 0 1 110 66 6067 0 0
1 116 66 6070 0 0 1 13 66 6089 0 0 1 336 66 6115 1 0 1 §Rent§ 0
0 2 70 66 6144 0 0 2 111 66 6149 0 0 2 111 66 6152 0 0 2 100 66
6156 0 0 2 13 66 6182 0 0 2 336 66 6201 1 0 2 §Food§ ...
```

The information recorded includes the language of the interface used, clock time the key was pressed, type of data that was entered (value, formula, label/text), row and column of the cell in which the key was pressed, the key pressed, and the final contents of the cell when the cursor leaves the cell. An example of a keystroke record is "**66 6201 1 0 2 §Food§**" taken from the raw data above is used to show the information recorded.

Language: 66 is the ascii number for B (for Bahasa Melayu)
Timestamp of keystroke: 6201 (ticks)
Type of contents: 1 for text
Column: 0 for Column A
Row: 2 for Row 3
Content of cell A3: Food

A program was written to convert the raw data above into a more readable form as shown below. The first column is the language of the interface, followed by the cell in which the keystroke was typed, the keystroke, and the time the keystroke was typed. The time shown below is the time that has elapsed since selecting the language to start using the spreadsheet.

```
@ ? Key pressed: b

?   A1      b           0 min. 0.000 sec.
B   A1      Started At:          0 min. 0.000 sec.
B   A1      _CR_        0 min. 9.451 sec.
B   A1      _DOWN_      0 min. 13.791 sec.
B   A2      R           0 min. 15.879 sec.
B   A2      e           0 min. 16.099 sec.
B   A2      n           0 min. 16.264 sec.
B   A2      t           0 min. 16.429 sec.
B   A2      _CR_        0 min. 17.473 sec.
B   A2      _DOWN_      0 min. 18.901 sec.
 A2Rent
B   A3      F           0 min. 20.495 sec.
B   A3      o           0 min. 20.769 sec.
B   A3      o           0 min. 20.934 sec.
B   A3      d           0 min. 21.154 sec.
B   A3      _CR_        0 min. 22.582 sec.
B   A3      _DOWN_      0 min. 23.626 sec.
 A3Food
B   A4      E           0 min. 25.275 sec.
```

C.4 Conventions Used in the Transcriptions

Conventions used were adopted from:

Ackerman, M., Starr, B., Hindus, D. and Mainwaring, S. (1997). Hanging on the Wire: A Field Study of An Audio-Only Media Space. *ACM Transactions on Computer-Human Interaction.* Vol. 4. No. 1. p39-66.

- { } Speech made by experimenter.
- [ ] Square brackets indicate external sounds or an inference on the part of the transcriptionists, data analyst, researcher. The inference in the square brackets [ ] will include extra information to provide a context for the speech.
- < > indicate a section of the tape that was indistinct.
- / \ and \ / indicates over-lapping conversational areas on the tape.
- = indicates that there is no appreciable pause between the two words.
- Three dots ... indicate a short pause; usually less than fifteen seconds.
- Longer pauses were indicated by a description of the pause, as in [pause 1.5 mins].
- Any text in italics (next to Bahasa Melayu words) refer to the English-equivalent of those words.
- An English translation of Bahasa Melayu words are in italics. Some words in Bahasa Melayu have different meanings depending on the context in which the words are used.
- Words are represented in a bold font to draw attention to them. For example, mis-pronounced words, negative comments, recurring theme/strategy like *trial and error* or underlined (English interspersed in Bahasa Melayu).
- Whole sentence is Bahasa Melayu are translated as a whole sentence. E.g. Masukkan sel pertama untuk diformat. *Enter the first cell to be formatted.*
- Words in English or similar to English are not translated. For example, Goto, Utiliti.

Representation of Arrow keys, Carriage Return, Function keys

_RT_	Right arrow
LEFT	Left arrow
_UP_	Up arrow
DOWN	Down arrow key
_CR_	Carriage return
_F1_	Function key F1

Keystroke Sequence

/ -Lajur-Lebar	Keystrokes pressed is /-L-L / is pressed to get the menu list. L to select command Lajur (*Column*) L to select command Lebar (*Width*).

C.5 Procedures to Calculate the SUS Scores

To obtain the System Usability Scale score, score contributions from all ten questions were totalled. Each score contribution ranged from 0 to 4. For questions 1, 3, 5, 7, and 9, the score contribution was the scale position minus 1; the scale position was 1 to 5, scale position 1 on the extreme left, and scale position 5 on the extreme right (Figure below illustrates the scale positions). For questions 2, 4, 6, 8 and 10, the score contribution was 5 minus the scale position (to compensate for reverse polarity). Once the total score contributions were obtained, the sum of the scores was multiplied by 2.5 to get the overall score of System Usability Scale. The justification behind the calculation procedures is available in Brooke (1996). According to Brooke (1998), products with good usability yielded SUS scores of 60 and above. Products with bad usability scored below 50.

1. I think that I would like to use this system frequently.

Strongly Disagree	Somewhat Disagree	Neither Agree nor Disagree	Somewhat Agree	Strongly Agree
1	2	3	4	5

System Usability Scale Question 1 (with Scale Positions 1 to 5)

C.6 Procedures to Determine Task Completion

There are three situations in which the participants are deemed to have not completed the task.

- Told: When participants were quite distressed, or appeared quite irritated, they were requested by the experimenter to move on to the next task.
- Exceeded "benchmark" time: If the participants spent more than 7 minutes on a single task, the task is considered complete. The time chosen takes into account that the participants may be exploring the spreadsheet to find the correct commands to complete the tasks. The time taken by each participant's attempt on each task was manually calculated based on the timestamps from the log files.
- Did not complete: Participants gave up, and did not complete the task.

Appendix D Profile of Participants

Participants	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15	P16	P17
Status	Low	Low	High	Low	High	High	Low	Low	Low	High	High	High	High	Low	High	High	High
Familiar	Yes	Yes	Yes	No	No	No	Yes	No	No	Yes	Yes	Yes	No	No	Yes	Yes	No
Gender	Fem.	Male	Male	Fem.	Fem.	Male	Fem.	Male	Fem.	Fem.	Fem.	Male	Fem.	Male	Male	Fem.	Male
Race	Malay	Malay	Non-Malay	Non-Malay	Malay	Malay	Malay	Malay	Malay	Malay	Malay	Malay	Malay	Malay	Non-Malay	Malay	Malay
Age	30	36	33	24	28	29	23	24	29	28	28	25	26	26	33	22	26
DOS Experience	No	Yes	Yes	Yes	No	Yes	No	Yes	No	Yes	Yes	Yes	Yes	No	Yes	Yes	No
Number of years used computers	1.5	14	13	2.5	4	11	3	4	4	9	13	10	15	2	11	12	10
Used Windows Excel 5.0	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Used Lotus 123? Dos/Win	W	DW	D	DW*	–	D	W3	–	W	D	D	D	D/W	–	D	–	W
Total spreadsheet use (hours)	> 40	>40	>40	>40	10-<20	>40	20-40	>40	>40	>40	>40	>40	>40	20-40	>40	>40	10-<20
Spreadsheet Use (hour/week)	>4	1-<2	1-<2	3-4	2-<3	<1	2-<3	3-4	>4	<1	>4	2-<3	>4	<1	<1	2->3	<2
Education Level[a]	Sec.	Sec.	MSc.	Sec.	Bach.	Bach.	Sec.	Sec.	Sec.	Bach.	Bach.	Bach.	Bach.	Sec.	MSc.	Bach.	Bach.
Degree in Computer Science	No	No	Yes	No	No	No	No	No	No	No	No	Yes	No	No	Yes	Yes	No
Language Preference (Bahasa Melayu/English)	BM	BM	Eng	BM	Bahasa Melayu	BM& Eng	BM	BM	BM	BM& Eng	BM	BM	BM& Eng	BM	Eng	Eng	BM& Eng
Speak Bahasa Sarawak[b]	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes
Understand Bahasa Sarawak	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes

Key: High Status Occupation: Manager, Lecturer, Systems Analyst, Tutor; Low Status Occupation: Assistant Accountant, Administrative Assistant, Clerks, Typist; Education Level[a]: Sec. (secondary school), Bach. (Bachelors, first degree); MSc. (Masters degree); Bahasa Sarawak[b]: Sarawak's version of Bahasa Melayu

Appendix E Number of Tasks Completed

Participants	Status	Fam.	Gender	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6	TaskTime Total	No. Task Completed
P12	High	Yes	Male	0:03:11	0:07:00	0:00:11	0:01:37	0:01:37	0:00:13	0:13:49	6
P4	Low	No	Female	0:02:46	0:02:30	(0:07:07[x])	0:01:07	0:00:36	0:00:18	0:14:24	5
P16	High	Yes	Female	0:01:40	0:01:42	(0:02:41[x])	0:00:53	0:03:16	0:00:29	0:10:41	5
P3	High	Yes	Male	0:04:06	(0:12:59[b])	0:00:28	0:00:36	0:00:38	0:00:30	0:19:17	5
P10	High	Yes	Female	0:02:10	(0:16:50[x])	0:00:26	0:01:35	0:00:48	0:00:24	0:22:13	5
P6	High	No	Male	0:03:12	(0:22:35[b])	0:00:27	0:01:36	0:00:44	0:00:05	0:28:39	5
P15	High	Yes	Male	0:06:28	(0:23:06[b])	0:04:02	0:05:25	0:03:48	0:00:15	0:43:04	5
P17	High	No	Male	0:02:35	(0:01:26[x])	0:03:32	(0:07:33[b])	0:04:03	0:00:21	0:19:30	4
P13	High	No	Female	0:02:40	(0:09:53[b])	0:01:13	0:02:11	(0:03:17[t])	0:00:42	0:19:56	4
P2	Low	Yes	Male	0:04:00	0:05:05	(0:01:05[x])	(0:13:07[t])	0:01:47	0:00:11	0:25:15	4
P5	High	No	Female	(0:10:14[b])	0:06:08	0:05:46	0:03:09	(0:18:28[x])	0:00:41	0:44:26	4
P1	Low	Yes	Female	0:03:08	(0:05:12[t])	–n/a-[a]	(0:15:11[b])	0:01:15	0:00:12	0:24:58	3
P8	Low	No	Male	0:03:57	(0:03:21[x])	0:00:50	(0:07:12[b])	(0:13:13[b])	0:00:16	0:28:49	3
P14	Low	No	Male	0:02:53	(0:07:46[t])	(0:02:24[x])	0:04:40	(0:10:40[x])	0:00:38	0:29:01	3
P11	High	Yes	Female	0:05:30	(0:09:03[t])	0:02:48	(0:11:03[t])	(07:56.0[t])	0:00:15	0:36:35	3
P9	Low	No	Female	0:06:08	(0:05:43[t])	(0:05:48[x])	(0:11:15[b])	(0:08:20[t])	0:00:24	0:37:38	2
P7	Low	Yes	Female	(0:11:47[t])	(0:02:48[t])	(0:03:21[x])	(0:10:25[t])	(03:41[t])	0:00:16	0:32:18	1

[a] Participant did not do the Save task; Participants did not complete the tasks because– [b]: exceeded benchmark time; [t]: told by experimenter; [x]: did not complete.

See Appendix C.6 for procedures to determine if the tasks have been completed.

Appendix F System Usability Scale Scores

The scores have been compensated for reverse polarity. Score of 5 indicates good usability rating.

PA	Status	Familiarity	Gender	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	SUS Score	Usability
P12	High	Yes	Male	5	5	4	5	5	4	5	5	5	5	95.0	Good
P3	High	Yes	Male	3	4	4	5	4	4	4	3	4	4	72.5	Good
P16	High	Yes	Female	3	4	2	5	4	4	3	4	4	4	67.5	Good
P5	High	No	Female	4	3	3	4	3	4	2	3	3	3	55.0	Neutral
P2	Low	Yes	Male	4	3	2	2	4	3	4	3	2	4	52.5	Neutral
P4	Low	No	Female	4	5	3	1	3	2	5	4	3	1	52.5	Neutral
P7	Low	Yes	Female	4	3	3	2	2	4	3	4	3	3	52.5	Neutral
P9	Low	No	Female	5	1	5	1	5	2	5	1	5	1	52.5	Neutral
P6	High	No	Male	1	2	2	5	4	4	1	4	4	2	47.5	Bad
P10	High	Yes	Female	2	5	1	2	4	3	5	2	4	1	47.5	Bad
P14	Low	No	Male	4	2	4	1	4	3	5	2	3	1	47.5	Bad
P1	Low	Yes	Female	2	2	3	4	2	2	2	4	2	4	42.5	Bad
P13	High	No	Female	2	5	4	2	1	5	1	3	2	1	40.0	Bad
P15	High	Yes	Male	3	5	4	1	2	2	3	2	2	2	40.0	Bad
P11	High	Yes	Female	1	5	1	2	2	2	1	4	2	4	35.0	Bad
P17	High	No	Male	1	5	3	4	1	3	1	1	1	1	27.5	Bad
P8	Low	No	Male	4	2	1	1	2	2	2	2	2	1	22.5	Bad
			Mean	3.06	3.59	2.88	2.76	3.06	3.12	3.06	3.00	3.00	2.47	50.0	

Grand Mean	50.00	Min.	22.5
Median	47.50	Max.	95.0
Stdev	17.05	Lower Quartile	40.00
SEMean	4.13	Upper Quartile	53.75

Appendix G Interview Responses of Participants

The rank used to order the responses in the table below is the Interview Rank.

	Rank	Sta-tus	Participants' Opinion about Spreadsheet
P12	1	High	*Bagus! Senang digunakan. Boleh komersilkan, saya rasa.* ("Good! Easy to use. Can be commercialised, I believe.")
P9	2	Low	*Ummm... hamparan itu bagus ya...* ("the spreadsheet is good...") untuk mmm... ("for umm...") menganti... mengantikan Excellah ("to replace... replace Excel-lah...") {*Mengapa?* ("Why")} Sebab dia ah... ("Because it... ah...") ah... *bagaimana nak cakap* ("how to say this...") [Snigger] lebih senang dari Excellah ("easier than Excel-lah.")
P8	3	Low	*Ahhh... saya rasa ia lebih* (" feel it is more")... effective if we use this.
P4	4	Low	I think it's okay, it's more, more like Lotus. I think we better use this in [department] there. ... I got the feeling it's much more better than Lotus, it's quite difficult at the first time. ...
P14	5	Low	*Saya rasa hamparan itu... saya rasa ia... kalau kita belajar.... ia mudah digunakan... {ummm} dan ia... senang untuk kita buat segala... < > yang kita naklah..* ("I feel that the spreadsheet... I feel it... if we learn... it's easier to use... and it... easier for use to complete all sorts of [tasks] that we want.")
P7	6	Low	*Ummm... Ia baik untuk kerja seharian... untuk menyimpan data...* ("It's good for daily work... and for saving data...") ... Umm... I think Excel *lebih senang lagi* ("Excel is easier")...
P16	7	High	For a beginner, it's quite difficult for them to learn this... to use this system {Okay...} Unless you are familiar with ahhh... other spreadsheets... then it should be very easy.
P5	8	High	*I think mudah guna Excel* ("I think easy to use Excel.") [Laughs] ... *Mudah lagi gunakan Excel* ("Easier to use Excel.") {Oh} *you nak... nak guna... Excel lebih mudah* ("You.. you want to use, Excel is easier.")
P17	9	High	\Spreadsheet/ I think it's really difficult to learn. Ahhhh... no, actually its not... not that difficult... [Negative comment, then mitigate the effects of critique] but I need to have this pre-testing ahh... {Ahh... okay} So that I... I can use it... because it's happened that I've used this thing quite some time before. I have to practise it... then I can... can make use of it. {Okay}
P15	10	High	I think the system still at a very primitive level... {ummm...} by providing some... umm... what do you call basic function {okay} and there are still a lot of room to improve the system... especially there are some inconsistency...
P3	11	High	Umm, old fashioned, umm, reminded me of Lotus 123.
P2	12.5	Low	...it's not that a bad utility ... it's not that difficult to use. You still have to go aa... looking for things lah, and have a trial and error. It's not that a bad utility to start the software to use the software now that it's not that difficult to use.
P6	12.5	High	Of course, if you compare with the... current spreadsheets we have in the market, of course that one is very outdated. ... Now, spreadsheet is is very advanced, ... So this one is very basic lah, in a sense. So, as a basic spreadsheet it's, it's okay.
P10	14	High	... I wanted to enjoy it but I was like ... ahhh shocked... because there's, like, nothing there.. nothing... kind of arahan ("commands") until I really got pening ("got a headache") ... I was taken aback... It wasn't that friendly I have to be frank with you. [Laughs] ...
P1	15	Low	Not user friendly.
P11	16	High	Menyukarkanlah... sebab saya tak biasa... dan... dulu saya biasalah, tapi... sekarang dah guna <> yang baru 'ni... dan banyak item-item yang tak ada... jadi... ini nama punya version yang <> Lotus one two three yang gunakan "straight" slash... ah... ah, sekarang dah guna... biasa gunakan mouse tik tik tik ... sekarang cari mouse tak jumpa mouse [Laughs] Dia kena, that mean dia kena baca dululah. *[Pause] Difficult... because I am not familiar... and... previously, I was familiar... ah... ah... now am using... accustomed to using a mouse tick tick tick (clicking noise of mouse)... now I look for mouse, I can't find the mouse. [Laughs] He/She needs, that means he/she needs to read first [probably referring to the instruction "Press / for list of commands"]*
P13	17	High	[Without hesitation] I don't like it. ...but this one is... it's easier but for someone who use to use Lotus it's difficult, you know? Okay?

Appendix H Usability Problems Isolated from Think Aloud Data

The usability problems described below were identified from the logging-augmented think aloud data collected during the evaluation of *Hamparan,* the Bahasa Melayu version of FIRST. All the usability problems are described together with the criteria used in identifying these problems, as well as examples/evidence of the usability problems. For information on the spreadsheet functionalities, refer to the TCALC User Guide in Appendix A. TCALC is the original spreadsheet with an American English interface.

Key:
The conventions used are shown below.

[P1:E3] Participant 1's Episode 3 in the think aloud transcript.

[P3:L] Participant 3's data from log file obtained during the thinking aloud session, that is, the sequence of keystrokes are shown. Keystroke sequence that appears below an Episode such as [P1:E3], means that the keystrokes are linked to that episode.

A keystroke sequence such "/"-hamparaN *Spreadsheet* and Simpankan *Save* means that the following keystrokes are pressed: "/" to select the list of commands, "N" to select the hamparan command, and "S" to select the *Simpankan* command. The English equivalent of the Bahasa Melayu commands are provided to aid readability of the keystroke sequence.

1. "/" Slash key was not clear.

Description:

The slash key is used to obtain the list of commands in *Hamparan* (similar to "/" key in Lotus 123). The instructions to obtain this list of commands is written on the top of the screen, that is, "Press / for the list of commands". Some participants, probably those who were unfamiliar with the Lotus-style spreadsheets, did not know how to get access to the "/" key.

Criteria:

The participant looked for a menu or commands that could help them complete the tasks such as increasing column width, saving the spreadsheet, changing the format or quitting from the spreadsheet. The participant "stumbled" or "wandered" around and seemed lost, looking for a way to complete the tasks.

Evidence:

Log file contains a number of arrow keys (cursor being moved around). After a lot of moving around or a long pause, the participants still had not use the "/" key and they verbalised that they could not complete the tasks. For example,

[P14:E16] *Setakat 'tu saja, tidak dapat dibesarkan* ("Only till here, [column] cannot be widened")

2. Used arrow keys to select the command

Description:

When "/" is pressed, the list of commands appears at the bottom of the screen. Each command is selected by pressing the capital letters in the text. For example, to select the Goto command, "G" is pressed. All the capital letters are in cyan while the remaining letters of the command are white. Instead of pressing the appropriate keys, a number of participants attempted to select the commands by using the right or the left arrow key, presumably to move the "cursor" to the required command.

Criteria:

Participant used "/" key and arrow keys and then followed by letters for commands which they had initially intended to choose.

Evidence:

Participant selected the Baris *Row* command and pressed the right arrow key to select the next command. And as the arrow key did not work, she chose Sisipkan *Insert* command.

[P5:L] /-B-_RT_-S ...

Participant P5 used the right arrow key to try to select command, and then selected 'B' for the Baris *Row* command, before choosing 'S' for the Sisipkan *Insert* command.

[P5:L] /_RT_-B-S-_CR_-_DOWN_

3. Confused between command and editing mode

Description:

The command mode is entered after "/" is pressed. Editing mode refers to the mode, which allows the user to edit the contents of the cell. Commands cannot be invoked in the editing mode. For example, if a user presses "/" to obtain a list of commands while in editing mode, the character "/"will be displayed in the cell. Some participants tried to edit cell contents in the command mode.

Criteria:

Participants were in editing mode when they pressed a character such as 'F' or a space and then tried to press "/" to get the commands. Sometimes the interaction is also accompanied by verbalised frustration.

Evidence:

In cell D5, participant P17 pressed "F" (which puts the spreadsheet in editing mode). The participant then tried "/" which did not bring up the command list since the spreadsheet was in editing mode. The participant felt frustrated (inferred from the rapid knocking on the table.)

[P17:E30] What else... what I'm trying to search... is the menu {Okay}
Menu, where's the menu...? [knocking on the table] Slash is not...
[working] Arh! [Frustrated]

[P17:L] F-/-_BS_-_BS_-/-_BS_-/-_CR_

4. Mixed-up between hamparaN *Spreadsheet* and Hapus *Delete* command

Description:

The participants were confused about the selection of the two menus as both menus begin with the letter "H". The two menus are Hamparan *spreadsheet* and Hapus *clear/delete.* Users have to press "N" to choose the hamparaN menu, which displays

commands like Load and Save, and "H" for the Hapus *Delete* menu, which deletes the contents of the spreadsheet. The capital letters in each command signifies the keys to press to select that command.

Criteria:

The participant verbalised that he or she wanted hamparaN but pressed "H" instead of "N" or participant chooses the Hamparan *Spreadsheet* command immediately after the Hapus *Delete* command had been pressed. For example, when the participant wanted to load a file using the Hamparan menu after just entering the spreadsheet.

Evidence:

Participant just entered into the spreadsheet and wanted to load a file using the hamparaN menu. Participant chose the Hapus *Delete* command when he actually wanted the hamparaN *Spreadsheet* command, as shown in the next sequence of keystroke.

[P2:E46] Alright *Hamparan* ("Spreadsheet") no, H should be N

[P2:L] /-Hapus-/-hamparaN-Muatkan *Load* -***-_CR_ [*** is the filename]

5. Expected more sub-commands

Description:

After the participants had selected / to show the list of commands, they expected to find more commands by pressing arrow keys or page down keys.

Criteria:

Participants verbalised the expectation, and this expectation is supported by key strokes such as /-_LT_-_RT_

Evidence:

Participant P6 expected more commands.

[P6:E14] {What's your expected ..} I expected that it will goes to the sub, ur, sub-command of the editing, sub-command of *suntingan* ("edit"). And it doesn't goes anywhere, where was it?

[P6:L] /-_LT_-_RT_

Participant P15 tried the Page-Down and the Down arrow key to find if there were other commands other that what is shown on the screen when / is pressed.

[P15:E59] Urr... I try it again... try the page down to get another set of urrr... instructions or down arrow seems that it's not working...

[P15:L] /-_PGDN_-_DOWN_...

6. Did not know how to exit from prompt/command mode

Description:

To exit from the prompt mode in the spreadsheet, the escape key is pressed. Some participants did not know about the escape key's function.

Criteria:

Participants are in the prompt mode and they pressed many combinations of keys to try and exit from the prompt mode.

Evidence:

Participant P16 was in the command mode (after pressing / key) and verbalised she did not know how to exit from the menu.

[P16:E41] Ahhh.... utiliti kah? *Hitung semula* ("Recalculate") *tunjukkan rumus* ("Show formula"). [Pause] Umm... no. Now where to... <> get out from helps [according to the participant, helps refers to the list of commands]...

[P16:L] /-Utiliti-Hitungkan Semula *Recalculate.* [Participant tries Utiliti and finds the options. From the log data, participant did not use ESC.]

7. Difficult to save the spreadsheet

Description:

The command sequence to Save the spreadsheet is: "/"-hamparaN *Spreadsheet* and Simpankan *Save.* Some participants had problems trying to save the spreadsheet.

Criteria:

Participants verbalised they wanted to save the spreadsheet and attempted to save the spreadsheet using different keys, for example, F1.

Evidence:

Participant P2 verbalised he wanted to save and pressed function keys F1, F2, F3 and F4.

[P2:E20-21] {Before that, suggest you save the spreadsheet} Uh huh, save save save save. Ah nope, {What are you thinking now?} Umm, trying to go somewhere to save. {<Just do the x>} Okay. This is totally different lah, I don't expect something like this.

Participant P16 also attempted to save the spreadsheet using the function keys, F1, F2, F3 and suggested ALT-F3.

[P16:E25] {Try to save this spreadsheet first} Umm... save... save... Alternate... F three, <is it?> How do you save.....

8. Message "Loading file..." displayed too briefly

Description:

To load a file, the user selects the following command sequence: /-hamparaN *spreadsheet* -Muatkan-**** where "****" is the filename. A message "Dalam proses memuatkan fail" *Loading file...* is displayed to indicate that the file is being loaded. However, the message is displayed too briefly, thus the participant is unable to read the message.

Criteria:

Participant verbalised this problem.

Evidence:

[P15:E62] There was a message... but I didn't see it clearly, the system... showing so fast and disappear like this...

9. Did not know meaning of Muatkan *Load*

Description:

To save a file, users choose the Muatkan *Load* command using the sequence: /-hamparaN *spreadsheet* -Muatkan-****. Soma participants do not know the meaning of Muatkan *Load*. This command is used to load the file onto the spreadsheet.

Criteria:

Participants verbalised they did not know the meaning of Muatkan *Load*.

Evidence:

After the participants chose N to select the command hamparaN *spreadsheet,* they verbalised that they do not know the meaning of *Muatkan*.

[P1:E22] Choose N... *Muatkan* ("Load")... what is this word? [Typing.]

[P5:E94] N... *Muatkan* ("Load") *Simpankan* ("Save...") *Apa maksud muatkan 'ni? Simpan... hamparan ya kan?* ("What is the meaning of this Load? Save... spreadsheet right?")

{*Awak faham maksud muatkan?* ("Do you understand the meaning of Muatkan [Load]?")}

Muatkan ("Load") meaning that's save *'aja kan* ("right")? ...

10. Did not understand the prompt "Masukkan sel pertama untuk diformat" *Enter the first cell you wish to format:*

Description:

This prompt appeared when the Format command is chosen. The correct input to this prompt is a cell address, that is, the first cell address of a cell-range. There is no indication what should be entered at the prompt.

Criteria:

The participant verbalised this problem or the participant typed in almost everything (except the cell address).

Evidence:

Participant P2 typed the following at the prompt.

[P2:L] /-Format-/-UP-BS-CR-CR-K-CR-CR-DEL-CR-DEL-CR-DEL-CR-CR-LT-BS-BS-DEL-DEL-UP-CR-DEL-UP-CR-DEL-LT-LT-LT-LT...

Instead of the cell address, participant P10 typed in formula at the prompt Masukkan sel pertama untuk diformat *Enter the first cell you wish to format*

[P10:E24] Umm... [Typing.] okay, mmm... B five *campur* ("add") ... C five ... D five ....

[P10:L] /-Format-B5+C5+D5...

11. Difficult to answer the prompt "Adakah anda inginkan semua sel diselaraskan ke kanan?" *Do you want the cell right-justified?*

Description:

This prompt appeared after the user had selected the Format command and entered the cell range. Participant did not know how to answer the prompt.

Criteria:

Participants verbalised their difficulty when they got "stuck".

Evidence:

When participant P13 was confronted with the prompt, she commented:

[P13:E35] *Adakah anda ingin semua sel diselaraskan ke kanan? Ke kanan?* ("Do you want all cells to be justified to the right? To the right?") Ahh... ah! *Ini tak ada yes no 'ni, bagaimana nak jawab?.* ("This has neither yes nor no, how to answer?") Enter? *Adakah anda ingin selaraskan < > ke kanan?* ("Do you want to justify to the right?")

12. Did not understand the prompt "Berapa tempat perpuluhan yang perlu dibundarkan untuk setiap nombor?" *How many decimal places should the number be rounded to?*

Description:

This prompt appeared after the user had selected the Format command, entered the cell range, chosen whether to right justify, and answered no to using commas for the numbers. Participants did not understand this prompt "Berapa tempat perpuluhan yang perlu dibundarkan untuk setiap nombor?" *How many decimal places should the number be rounded to?*

Criteria:

Participants verbalised their lack of understanding..

Evidence:

When participant P12 was faced with the prompt, he said:

[P12:E77] *Perpuluhan mmm, tak ada. Macam mana? Tiada? Tiada?* ("Decimal, don't have. How [to do this]? Don't have? Don't have?")

13. Mistook Hapus *Delete* for deleting rows or columns

Description:

The Hapus *Delete* command deleted the contents of the cell where the cursor was in. Participants used the Hapus *Delete* command to delete rows or columns instead of using the keystroke sequence /-Baris *Row*-Hapuskan *Delete* or /-Column-Delete.

Criteria:

Participants accidentally inserted a column or row and to remove the rows or columns, instead of the /-*Column*-*Delete*, they used /-Hapuskan *Delete* command

Evidence:

Participant P8 accidentally inserted a column. He wanted to delete the column but used the /-Hapus *Delete* command

[P8:E53] Hapus *Delete*

[P8:L] /-Lajur *Column*-Sisipkan *Insert* [followed by] /-Hapus *Delete*

Realising his mistake, P8 used the correct command to delete the inserted column.

[P8:E54] L Lajur *Column*, hapuskan *Delete*

[P8:L] /-Lajur *Column*-Hapus *Delete*

14. Did not know meaning of Goto

Description:

The command Goto is on the list of commands of the spreadsheet. The command is used to go to a particular cell. The participants did not understand what Goto meant.

Criteria:

Participants verbalised that they did not understand Goto command.

Evidence:

[P5:E22] ... *Gotor I tak faham?* ("Gotor, I don't understand.") ...

[P11:E36] ... Goto... Goto *'ni buat apa* ("What does goto do?") ..

15. Did not understand the Goto prompt "Masukkan sel yang diingini?" *Enter cell to go to?*

Description:

When the user selected the Goto command, the user is prompted with "Masukkan sel yang diingini" *Enter cell to go to* Valid input for this prompt is the cell address that the user wants to go to.

Criteria:

Participants typed in something other than the cell address.

Evidence:

Participant P9 did not understand the Goto prompt and entered A+b at the prompt.

[P9:E52] *Masukkan sel yang diingini* ("Enter the cell you want [to go to]")...
Yes... [Typing.] [Beep!] ...

[P9:L] /-Goto-A+b

Participant P5 typed in RM at the Goto prompt.

[P5:E77] Goto... Goto <> goto...

[P5:L] /-Goto-RM_CR_

16. Did not know meaning of Lajur *Column*

Description:

When "/" is pressed, Lajur Column is a command available from the list of commands in the spreadsheet. Participants did not know the meaning of Lajur Column.

Criteria:

The participants verbalised that they did not know the meaning of lajur *Column.*

Evidence:

[P17:E38] *Masukkan* ("Load") *Hapuskan* ("Delete")... What is *Lajur* ("Column")?

[P12:E5] What is *Lajur* ("Column")? I think....

17. Unit of column width was not known

Description:

To change the column width, an integer has to be entered at the prompt "Taipkan lebar lajur baru" *Enter the new column width.* That unit corresponds to the number of characters in the word (or value) to be placed in a cell.

Criteria:

Participants, who were unaware of the unit accepted as input. They entered a number but were unsure of the unit. For example, a participant typed in 1 " or 1 inch for the column width. Participants also verbalised their difficulty.

Evidence:

Participant P6 described his difficulty.

[P6:E83] Uhh, okay, I wonder why the system do not show what is the current size of the lajur. Now I don't know what, what aaa... figure to type in, because I don't know what's, what's the unit the system is using to... to... to measure the size of the *lajur* ("column")... the size of the column.

Participant P5 did not know the unit of the column-width and tried ' and " which are abbreviations for feet and inches. P5 also typed "IN" for inch.

[P5:E52] *Lajur baris* ("Column Row")... L... L... *Taipkan lebar lajur* baru ("Type the new column-width.") *Lebar dia... dia berapa?* ("The column-width... How big is the column-width?") [Pause] *Lajur lama berapa?* ("The old column-width is how big?") [Laughs] *Lajur lama berapa* ("How big is the old column-width?...") *Tak tahu* ("Don't know.") [Laughs] *Apa taipkan lebar lajur baru*? ("What is [the prompt] type the new column-width?") {*Apakah yang diingini?* What is required?*} Ahh, yalah* ("yes")... *Berapa lebar dia kan?* ("What is the column-width?") {Ahh...} Ummm... *Tigalah... inci...* ("Three... inches...") Umm... inch... Umm? [cannot enter?]

[P5:L] /-Lajur *Column*-Lebar *Width*-3-'-"-IN-_CR_

18. Did not understand the prompt "Taipkan lebar lajur baru" *Enter the new column width*

Description:

The participant typed anything at the prompt, except a number. If they typed in a number, then the participants most likely understood the meaning of the prompt but were unsure of the unit of the column width.

Criteria:

Participant typed in any combination of characters except an integer.

Evidence:

Participant P1 typed in a cell address at the prompt.

[P1:E23] ... try slash again. Ah, we... have to choose *Lanjur* ("column")

Sisipkan ("Insert")... okay *Lebar* ("Width")... choose A six. [Typing.] [Pause]

[P1:L] /-Lajur *Column*-Lebar *Width*-a6Aa-_ESC_

Participant P7 typed in "eeE" probably the word "electricity"; the column-width was to be widened to take in this word.

[P7:L] /-Lajur *Column* -Lebar *Width*-eeE-_CR_

19. Did not know meaning of Baris *Row*

Description:

Baris *Row* is a command available, when "/" is pressed, from the list of commands in the spreadsheet. Participants did not know the meaning of Baris *Row*.

Criteria:

Participants verbalised that they did not understand the Baris command.

Evidence:.

[P5:E22] *Baris Row... Baris 'tu maksudnya apa?* ("What does Row mean?") *Tak faham.* ("Don't understand.")

20. Did not know meaning of Sisipkan *Insert*

Description:

When /-Baris *Row* or /-Lajur *Column* is pressed, the participants are provided with the Sisipkan *Insert* command. Participants did not know the meaning of Sisipkan *Insert*.

Criteria:

Participants thought "sisipkan" was a command other than insert, or they verbalised that the did not know what the command meant.

Evidence:

Participant thought sisipkan *insert* would calculate the total of the cells.

[P14:E58] *Baris*, ("Row") S... Oh... {What do you expect to do... I mean, what do you expect *Sisipkan Insert* to do?} Sisip *Insert*? Ah... *saya nak untuk mendapatkan jumlah* ("I want to get the total...") {Ahh hah} *Jadi saya* I... *tekan Sisip* ("So... I press Insert.")

Participant did not know the meaning of sisipkan *insert*.

[P6:E37] I'm trying to see how ... what does *sisipkan* ("Insert") means?

21. Confused between Baris *Row* and Lajur *Column*

Description:

Participants mistook Baris *row* with Lajur *Column* commands.

Criteria:

The participants chose the Baris *Row* command when they actually wanted the Lajur *Column* command.

Evidence:

Participant P15 was attempting to complete the column-width task. He initially used the Baris *Row* command, and then later found the Lajur *Column* command which was the correct command to use.

[P15:E85] Try another option... Maybe its under Baris *Row* ... No.

[P15:E90] Now I'm lost... [Pause] Now I have to try again with the option <> under menu... Try the *Layjor* ("column")... Oh.. I think I made a mistake just now. I should try this earlier, it seems that there is an option for me... on the *lebar* ("Width")... Okay, so I found it... ...

22. Did not know meaning of Sunting *Edit*

Description:

Another command that is available from the list of commands is the Sunting *Edit* command. Some participants did not know the meaning of Sunting *Edit.*

Criteria:

The participants provided a different meaning for the word sunting *edit,* or the participant verbalised he or she did not know what Sunting *Edit* meant.

Evidence:

Participant P4 associated the meaning of Sunting *Edit* with film editing.

[P4:E16] {Can you guess the meaning of *sunting* ("Edit")?} *Sunting... filem yang kena suntingkan.* ("Edit... film that gets edited") [Laughs]

[P11:E50] What is this one *sunting* ("edit")?

23. No feedback to Sunting *Edit*

Description:

When the participants selected /-Sunting *Edit,* they did not know they were in the editing mode. There was no feedback to indicate the spreadsheet was in edit mode.

Criteria:

Participants pressed S to select Sunting *Edit,* and then as nothing happened, they pressed S again.

Evidence:

Participant P9 selected Sunting *Edit* and then pressed S again as she thought that nothing had happened. Once Sunting *Edit* is selected, the spreadsheet is in edit mode.

[P9:L] /-_RT_-Sunting-s

After selecting S, Participant P11 thought the Sunting *Edit* command did not work.

[P11:E50] Oh *sunting* ("edit"), *sunting tak ada* ("edit doesn't 'work'")?

24. Did not know Utiliti command

Description:

Participants did not know the meaning of the Utiliti command, a command in the list of commands. Under the Utiliti menu item, is the Tunjukkan Rumus *Formula Display* and Hitung Semula *Recalculate* command.

Criteria:

Participant associated a different meaning with Utiliti or did not know what utiliti is for.

Evidence:

Participant P4 associated Utiliti with electric (utility) bills.

[P4:E16] {How about utiliti?} Utiliti at ***, utiliti, we use for *bil elektrik.* ("electric bills"). ...

Participant P14 did not know the meaning of Utiliti

[P14:E44] Utiliti, *apa makna* ("what is the meaning of") utiliti...

25. No feedback to Hitung Semula *Recalculate*

Description:

If the Automatic Recalculation mode is off, formulae will not be recalculated if any values, referred by the cell addresses in the formula, are changed. When Automatic Recalculation is off, the Hitung Semula *Recalculate* command is used to update the changes. Automatic Recalculation is toggled on or off by using the key sequence /-Auto. When users choose the Recalculate command, from the Utiliti command (and the Automatic recalculation is off), there is no feedback as to whether the recalculation has worked or not.

Criteria:

After selecting the Hitung Semula *Recalculate* command, it seemed to the participant that nothing had happened.

Evidence:

After selecting Hitung Semula *Recalculate,* participants P3 and P10 commented:

[P3:E26] You... what is *Hitung Semula* ("Recalculate")? Didn't do anything.

[P10:E39] Umm, eh,... how come there's nothing...

26. Hitung Semula *Recalculate* command was not intuitive

Description:

The Hitung Semula *Recalculate* is a sub-command found under the Utiliti command. The Hitung Semula command carries out the recalculation when the Automatic Recalculation mode is off.

Participants did not know what this command did.

Criteria:

Participant kept on using this command for the wrong tasks, as the participants did not know what the command did.

Evidence:

Participant P8 used the Hitung Semula *Recalculate* command to change the currency format:

[P8:E83] I have to.... change the total to currency format. So... I have to... *hitung semula* ("Recalculate")

Participant P10 thought the Hitung Semula *Recalculate* was for creating a formula, thus she expected a prompt to ask for a cell address which is to be added as part/component of a formula.

[P10:E69] Ah, < > utiliti *hitung semula* ("Recalculate") *tunjukkan* utiliti ("Show utility") So if I was thinking, so if you [were] to *hitung semula* ("Recalculate"), then the next *arahan* ("Instruction") to be ah... ah next *arahan* ("Instruction") is going to ask you... which one to add ... I was expecting that. *Tapi tak ada* ("but do not have")...

27. Tunjukkan Rumus *Formula Display* mode was not obvious

Description:

When the Tunjukkan Rumus *Formula Display* mode is off, numbers/values will be displayed in the cells where formulae are present. When the mode is on, the formulae in the cells are shown. Participants who accidentally turned the Tunjukkan Rumus *Formula Display* mode on, were confused when the formulae (they entered) were displayed as formulae even though they could see at the status bar that their formulae were correct.

Criteria:

Participants did not know the Tunjukkan Rumus *Formula Display* mode was on (which they had inadvertently toggled on).

Evidence:

Participant P6 had turned on the Formula Display mode. Thus, when he entered the correct formula, he did not know why the formula was in the cell but not the answer (number).

[P6:E49] Ohh, it shows the answer on the bottom LT corner, but the answer doesn't go into the b five cell where it should be displayed. I wonder why it worked like that.

After entering the correct formula for the total, participant P12 also wondered why the answer did not appear in the cell or as the participant put it "cannot print it out here".

[P12:E33] Yes, forty five, oh, this is the answer. Four eight five. But how how come I cannot print it out here? ...

28. Confused with /-Utiliti-Tunjukkan Rumus *Show Formula* and /-Utiliti-Hitung Semula *Recalculate*

Description:

Participants were confused with the Tunjukkan Rumus *Formula Display* command and the Hitung Semula *Recalculate* command.

Criteria:

Participant chose one command although verbalised choosing the other command.

Evidence:

Participant P12 managed to switch off the Show Formula mode i.e. the value was shown in the cell instead of the formula (using the Tunjukkan Rumus *Formula Display* command). However, when participant P12 went back to switch on the Tunjukkan Rumus *Formula Display* mode, he chose the wrong command i.e. he chose the Hitung Semula *Recalculate* command. He tried using the Hitung Semula *Recalculate* command four times before he realised it was the wrong command to use.

[P12:E90] Oh, I think I know. [Typing.] Ah! [Got it! Eureka] Hah! Oh! [laughs] Oh! I this is good. Oh! Now I know.

[P12:L] /-Utiliti-*Show Formula*

[P12:E91-96] Hah! Try again. Should be utiliti, hitung semula *Recalculate*, Oh! Hitung semula is.. ah.. okay.. Utiliti, tunjukkan formula... *Show Formula*... Ah! Ya, ya ya ya yes. *Yes.* Mmm... utiliti, eh? U untuk *for* utiliti, uhhh, for the < > H. Eh? Why nothing happen? [Typing.] U, utiliti *hitung semula.* ("Recalculate"). {What did you expect to happen?} Ah, it should come up with four eight five. How come ah? Just now, I could can make it to-ooo... you know.

[P12:L] /-Utiliti-Hitung Semula Recalculate
/-Utiliti-Hitung Semula Recalculate
/-Utiliti-Hitung Semula Recalculate
/-Utiliti-Hitung Semula Recalculate

[P12:E91-98] Try again.. utiliti. Tunjukkan rumus *Show Formula* Oh! [Comprehend]. I think I know switch it on and switch it off. Okay! Kay kay okay! Finish!

[P12:L] /-Utiliti-*Show Formula*

29. Tunjukkan Rumus Formula Display command was not intuitive

Description:

The participant did not know what the command Tunjukkan Rumus *Formula Display* does.

Criteria:

Participants verbalised that the command did not do what they expected.

Evidence:

[P6:E22] Ahh utiliti, *Tunjukkan Rumus* ("Formula Display"). I'm expecting the system to show... the various formula options to choose, and it doesn't do so.

Participant P8 thought the command had something to do with the currency format task i.e. to change the format of the cell to RM format (RM is the currency symbol for Malaysian Ringgit).

[P8:E105] [Typing.] Uhhh, utiliti I think so lah. Tunjukkan Rumus *Formula Display*. RM... Not this one.

30. Did not know meaning of Auto

Description:

Automatic Recalculation (or Auto) is a command in the spreadsheet which controls the automatic recalculation of formulae in the spreadsheet. If the Automatic Recalculation mode is on, the spreadsheet will automatically recalculate formulae every time a change is made in the spreadsheet. If the mode is off, then the formulae can only be recalculated using the command sequence /-Utiliti-Hitung Semula *Recalculate.*

Criteria:

Participants verbalised they did not know the meaning of Auto command.

Evidence:

[P1:E93] What is auto? Try the auto.

[P4:E19] Let's try this out. Auto, what could Auto be?

31. Feedback of Automatic Recalculation was not obvious

Description:

If the Automatic Recalculation mode is on, the text "PerhitunganAuto" (abbreviation of Perhitungan Automatik *Automatic Recalculation*) will appear at the top right corner of the screen. If the mode is off, the text PerhitunganAuto will not be displayed at the top right corner. Participants were unaware of this mode change.

Criteria:

Participants select the Auto command twice. The second selection was made as the participants thought nothing had happened (participants did not see the "Perhitungan Auto text"), thus made the selection again. This reselection could be seen in the log file in which the participants tried the Auto command twice.

Evidence:

Participant P15 tried Auto command, initially typing a lowercase letter "a". As the participant did not see any difference, the participant tried Auto again, this time using the capital "A" to select the commands.

[P15:E39] Maybe I can try Auto... I have to understand what auto means... Typing]

[P15:L] /-Auto [using a to select Auto]
/-Auto [using A to select Auto]

Participant P9 selected the Auto command, and typed in "a" again to reselect the auto command as she probably did not notice any changes after the first selection.

[P9:L] /-Auto [using a to select Auto]
/-Auto [using a to select Auto]

32. Thought Keluar *Exit* was to Open/Load (in Bahasa Melayu keluar also means "take/bring out")

Description:

The command sequence /-Keluar *Exit* is used to quit from the spreadsheet. However, some participants misinterpreted the word Keluar *Exit*.

Criteria:

The participant associated the word "Keluar" with "to take/bring out" as opposed to exit.

Evidence:

Participant P14 thought that the keluar *quit* command was to open the spreadsheet.

[P14:E33] *Nak... Kita nak keluarkan, nak dapatkan... yang ditaip tadi?* ("We want to take out, get out... [the spreadsheet] we typed just now?") {*Ya* ("Yes")} Ah.... [Typing] *Ya* ("Yes")... okay...

[P14:L] /-Keluar *Quit*

33. Unable to Keluar *Quit*

Description:

Some participants were initially unable to quit as they were unable to find the command to quit from the spreadsheet.

Criteria:

To quit, the participants have to choose the hamparaN *spreadsheet* menu. In most software applications such as Word and Excel, the Exit command is in the first (File) menu. Participant probably thought the *Keluar* command was in the first menu. Only when they found that the Quit command was not there, the participants tried / to get the command list to find the quit command. The participants found the Keluar *Quit* command the second time round.

Evidence:

Participant P3 tried the hamparaN *spreadsheet* menu only to find the quit command was not there. Participant then "escaped" from the hamparan *spreadsheet* menu and found the Keluar *Quit* command.

[P3:E56-57] Why do you have...? Let's see There's nothing else you can do. Slash ahhh... [found it] *Keluar* ("Quit")

[P3:L] /-hamparaN-ESC
/-Keluar Quit -T[idak No]

Participant P13 initially tried hamparaN *Spreadsheet* menu, read the commands in the first menu list and found that Quit was not among the commands. In the second attempt, the participant found the Quit command.

[P13:E69] Slash, ahhh... *keluar dari hamparan* ("Exit from the spreadsheet,") N... enter *Simpankan, cetakkan, hapuskan...* ("Save, Print, Delete...")

[P13:73] [Heh heh, snigger], Slash, *sekejap, ahh...* ("Slash, just a minute, ahh...") Format, *hapus, goto, keluar oh keluar K* ("Format delete, goto quit... oh quit K...") *hamparankah? Keluar. Dah!* ("Spreadsheet? Quit. Finish!")

[P13:L] /-hamparaN *Spreadsheet* - _ESC_
/-Keluar *Quit*

34. Need to press Enter or other arrow key to go to next cell

Description:

After a label or value has been typed into a cell, the users have to press the Enter key or Carriage Return (CR) to enter the contents into the cell. After the CR has been pressed, the cursor will remain at the cell address into which the data has been entered. The users would have to press an arrow key to go to the next cell. Unlike, in Excel, users can use either the CR, Enter, Tab or arrow key to enter the data and move the cursor to the next cell. Also, right arrow keys do not enter data into cells in *Hamparan.*

Criteria:

Typing arrow key is not enough to enter the data, and move to the next cell. Participants have to use CR, and then arrow key to move next cell.

Evidence:

Participant P1 was correcting the word January in cell B3. When she completed the correction, she used the _RT_ (right) arrow key to enter the data. When this did not work, she used the CR key.

[P1:L] ... Januaru-_BS_-y-_RT_-_CR_-_RT_ ...

Participant P1 entered 200 into cell B4, and used _RT_ arrow to enter the data. As this did not work, she used _CR_ to enter the data.

[P1:L] ... 200-_RT_-_CR_-_DOWN_

35. Formula syntax was not intuitive

Description:

The word "intuitive" here means intuitive to participants who have used spreadsheets before. As the participants have had no experience using the FIRST spreadsheet, the formula syntax in spreadsheet were not known to them. The correct formulae to sum cells B2, B3 and B4 are B2+B3+B4 and B2:B4.

Criteria:

Participants commented they are unable to complete the formula or participants tried several types of formulae.

Evidence:

[P1:L] @SUM(B4+B5+B6)

[P3:L] =sum(b2..b4)

[P15:L] =TOTAL(B2..B4)

36. Increasing column width was not straightforward

Description:

To increase the column width, the users select /- Lajur *Column* -Lebar *Width.* Some participants did not know what commands to use to widen the column width.

Criteria:

Participants verbalised their difficulty in completing the task. Participants also used other commands to complete this task.

Evidence:

Participant P17, in episodes 34 to 43, tried a number of commands to complete the task, see below. Participant P17 took 26 attempts to complete this task.

[P17:L] /-Format-_ESC_

/-Utiliti-_ESC_

/-Baris-_ESC_

/-Lajur-_ESC_

/-Sunting-_CR_

/-Sunting

/-Utility-_ESC_

/-Auto-_ESC_

/-Format-_ESC_

/-hamparaN-_ESC_ ...

/-Lajur Column-Lebar Width-14_CR_

/-Lajur Column-Lebar Width-12_CR_

Participant P15 also tried a number of commands to complete this task. He took 10 attempts to complete this task. Below are attempts made by participant P15 to complete this task.

[P15:I] /-Baris Row-_ESC_

/-Auto

/-Format-A4-A4-y-no-y-n-y

/-Utiliti-_ESC_

/-Baris Row-_ESC_

/-hamparaN Spreadsheet -_ESC_

/-Utiliti-_ESC_

CTRL-_RT_

Tried shift arrow keys

/-Lajur Column-Lebar Width-11_CR_

37. Navigation using tabs

Description:

In Excel, participants can use the TAB key to move the cursor to the adjoining cell on the right, that is, navigate to the next cell. In *Hamparan*, TAB key does not do anything.

Criteria:

Participants attempted to use TAB keys to navigate in the spreadsheet.

Evidence:

Participant P6 tried to use TAB to enter the data and move to the next cell when entering the label "JANUARY" and "FEBRUARY" in cells B1 and C1 respectively.

[P6:L] ...JANUARY-_TAB_-_CR_-_RT_-FEBRUARY-_CR_-_TAB_-_RT_ ...

Participant P7 in cell A1, tried to enter Rent and then used TAB to move to the next cell. The TAB did not enter the data into the cell nor move to the next cell on the right.

[P7:L] Remt-_BS_-_BS_-[Pressed TABs five times]-[Pressed _BS_backspace four times to delete Rent]-TAB-_RT_-

38. Expected carriage return to move to the next cell

Description:

In Excel, typing the carriage return (CR) key will do two things: (i). enter the data into the cell the cursor is currently on, and (ii). move the cursor down to the next cell. In *Hamparan*, CR enters the data but does not move to the next cell. The users have to use arrow keys to move to the next cell. Some participants had problems adapting to use this CR.

Criteria:

Some participants entered the data, typed CR and started typing the next data. What they inadvertently do is type the next cell's data into the same cell as CR key did not move the cursor to the next cell.

Evidence:

Participant P5 typed the row heading "RENT" and pressed DOWN arrow key in cell A1. Participant also typed "FOOD" in A2, pressed CR to enter the data, but did not press arrow keys to move to the next cell. Participant typed "ELECTRICITY" which is still entered into A2. Participant realised that she entered "ELECTRICITY" into the wrong cell.

[P5:E29] Ohh... *salah* ("wrong") [made a mistake]. [Typing]

Participant P5 retyped "FOOD" into A2, and "ELECTRICITY" in A3, "TOTAL" is entered into A4.

[P5:L] ... [In cell A1] RENT-_CR_-_DOWN_

[In cell A2] FOOD-_CR_-ELECTRICITY-_CR_-_DOWN_

[In cell A3] -_UP_-

[In cell A2] -_DEL_-FOOD-_CR_-_DOWN_-

[In cell A3]-_ELECTRICITY_ ...

Participant P3 pressed CR and discovered the CR behaviour in *Hamparan*. He commented:

[P3:E10] ... Oh, cannot go to next line that's all is it?

39. Confused between editing mode and navigation mode

Description:

Hamparan enters editing mode when a user types a character (other than "/" or CR in the spreadsheet). Participants try to navigate to other cells without realising they are in the editing mode. (Right and left arrow keys will not move the cursor, though UP and DOWN will function as a CR).

Criteria:

Participants got into the command mode and then used arrow keys to try and navigate to other cells.

Evidence:

Participant P9 typed "J" for January. She deleted the character and pressed the arrow keys to try and move the cursor which she could not (as *Hamparan* is in editing mode).

[P9:E2] Oh [understood that she could use the arrow keys to navigate.]

[P9:L] [in cell A1] J-_BS_-_UP_-UP_-_LEFT_-_UP_-

[In cell B1] _RT_-_LEFT_

Participant P13 got "stuck" after pressing the space key, which entered the spreadsheet into editing mode. She tried the RIGHT arrow and LEFT arrow keys, and ultimately pressed UP which acted as a CR. Participant P13 verbalised her frustration.

[P13:E29] ... Susahlah this thing. Ummh, ummh, its difficult to get ummm...

susahlah nak dapat ... ("Difficult to get...") ...

[P13:L] _ _[Space]-RT_-_RT_-_RT_-_RT_-_RT_-_LEFT_-_RT_-

_RT_-_RT_-_RT_-_LEFT_-_UP_-_RT_- [in cell B5]

40. Unexpected exit sequence

Description:

An error in the software caused the software to quit (not crash) when /-_LEFT_ is pressed. The normal command sequence to quit is /-Keluar *Quit-...* (This keystroke sequence is undocumented.)

Criteria:

Participants used the /-_LT_ sequence.

Evidence:

Participant P2's command sequence of /-_LEFT_ resulted in the exit from the spreadsheet. The final (undocumented) key combination SHIFT-F1 acted like an ESCAPE key which quit the spreadsheet without saving.

[P2:E34] Oh oh {okay } {Crash} It's not there. [Laughs]

[P2:L] /-_DOWN_-_DOWN_-_LEFT_-_LEFT_-...-SHIFT-F1

Participant P4 also discovered the exit sequence of /-_LEFT_. Participant P4 initially typed her name as the filename. Realising the prompt "Simpan hamparan ini?" *Save this spreadsheet?* required a yes/no answer, she entered "y" and typed in her name as her filename.

[P4:L] /-_LEFT_-***-y-***-_CR_

41. Crashed at Formatting

Description:

In Excel, formatting can be carried out in a cell range that cuts across different rows or columns. *Hamparan* allows only a cell range that comprises cells in the same row or same column.

Criteria:

Participants crashed the spreadsheet when they input a cell range that is not in the same row or same column.

Evidence:

Participant P15 attempted to change the currency format for the cells in the range B2 to D5 which caused the system to crash.

[P15:E93] So the system does not changing the format to Ringgit Malaysia format. And I have to try again with the format and repeat the same process... by selecting... by telling the system that the range I am supposed to key in.

[P15:E94] [Beep!] Oh... the system.... hung.... The system quit. [Laughs]

[P15:L] /-Format-B2-D5-_CR_

42. Unexpected crash at Column-Width

Description:

A software problem causes *Hamparan* to crash when the users entered a CR, DOWN or a value outside the acceptable range at the column-width prompt. For example, crashes will occur with these the command sequences: /-Lajur *Column*-Lebar *Width*-_DOWN_ or /-Lajur *Column*-Lebar *Width*-_CR_ or /-Lajur *Column*-Lebar *Width*-x-_CR_ (where x is a value less than 3, and greater than 77).

Criteria:

Participants crashed the spreadsheet (while completing the column-width task) using the sequence /-Lajur *Column*-Lebar *Width*-_DOWN_ or /-Lajur *Column*-Lebar *Width*-_CR_ or entering a value which is outside the accepted column-width range of the spreadsheet.

Evidence:

Participant P8 crashed the spreadsheet after entering the following sequence:

[P8:E37] [Beep] Osh... [Crash]

[P8:L] /-Lajur *Column*-Lebar *Width*-_CR_ [Crash]

Participant P11 crashed when she entered a 2 for the column-width.

[P11:E45] *Saya pergi lajur, saya lebarkan dia...* ("I go to column, I increase the column-width...") *Taipkan lebar lajur baru, dua.* ("Type the new column-width, two.") [Crash]

[P11:L] /-Lajur *Column*-Lebar *Width*-2 [Crash]

43. Unexpected load sequence /-hamparaN-_RT_

Description:

The command sequence to load a file in *Hamparan* is /-hamparaN *Spreadsheet*-Muatkan *Load*-<filename>-_CR_. An unexpected (undocumented) sequence to load the file was discovered by the participant.

Criteria:

An undocumented way to load file was used by participant i.e. /-hamparaN *Spreadsheet*-_RT_-

Evidence:

Participant P7 pressed /-hamparaN *Spreadsheet* and _RT_ (which appears to be equivalent to load a file, a software error). The participant was prompted whether to save the file. The participant answered Y(a) *Yes* correctly and saved the file as the participant's name. The participant was prompted for the filename to be loaded.

[P7:L] /-hamparaN-_RT_-ya *yes-filename*-_CR ...

44. Treated DOWN and UP arrow key as CR key

Description:

Participants used the UP and DOWN arrow key to enter the data. The arrow keys do not move the cursor to the next cell which is inconsistent with other spreadsheets such as Excel. Also, to add to this inconsistency, only the UP and DOWN arrow keys enter the data but not the LEFT and RIGHT key. This inconsistency may cause confusion when participants use the various keys to enter data.

Criteria:

Participants used the UP and DOWN arrow key to enter data. And they also use RIGHT or LEFT arrow keys to enter the data.

Evidence:

Participant P12 used the DOWN key, which he thought would enter the data, and moved the cursor down to the next cell. The DOWN key entered the data but did not move the cursor down. As the cursor did not go down to the next cell, and he had typed "Electricity" into the cell, he deleted "Electricity"...

[P12:L] FOOD-_DOWN_-Electricity-/-Hapus *delete*-

and retyped FOOD. This time he used DOWN-DOWN to enter the data.

[P12:L] FOOD-_DOWN_-DOWN_

After he typed "Electricity", he then used CR down to enter the data.

[P12:L] Electricity-_CR_-_DOWN_

After typing "TOTAL", he tried to use the RT arrow key to enter the data and move to the right (which did not work). Then, he used CR.

[P12:L] TOTAL-_RT_-_CR_-_UP_

Later he used the DOWN DOWN key combination again to enter the data.

[P12:L] 200.00-_DOWN_-_DOWN_

45. Could not delete

Description:

This is a program error. When the complete contents of a cell are deleted using edit or F2, and a CR is entered, the contents will not be deleted. The carriage return (CR) appears to act like an Escape key in this case.

Criteria:

Participants tried to delete the formula using F2 (enter editing mode) and backspaced to delete all the characters in the cell. However, when CR is pressed to enter the "deletion" into the cell, the characters were not deleted. This non-deletion only occurs when an attempt to delete all contents of the cell is made.

Evidence:

Participant P1 wanted to delete the incorrect formula @SUM(B4+B5+B6). She used function key F2 and used backspaces to delete the formula. When she hit the CR key, the formula was still in the cell.

[P1:L] _F2_-[_BS_ 14 times to delete the formula]-_CR_-

46. Had to translate Bahasa Melayu terms to English

Description:

The Bahasa Melayu commands or words were translated to English before the participants proceeded with the tasks. As the participants are familiar with applications in English, some participants translated the prompts to English.

Criteria:

Participants verbalised the translation of Bahasa Melayu commands into English.

Evidence:

Participant P13 was attempting to save the spreadsheet (translation in bold):

[P13:E41] Oh, *simpankan hamparan anda.* ("Save your spreadsheet") Okay, hamparan ...*simpankan hamparan, mana nak* **save** *ni?* [Knocking on the table] ("Spreadsheet... save spreadsheet, how to save?") **Save,** *ya, ya, ya,* ("Yes yes yes").

Participant P16 was using the format command:

[P16:E43-44] Format. *Masukkan sel pertama untuk diformat.* ("Enter the first cell to format.") *Sel pertama* ("First cell.") **First cell... to be formatted** [Translate to English]

47. Not sure how to answer yes or no or ya *yes* or tidak *no*

Description:

At some prompts, the participants are required to give a yes/no answer. However, some participants were unsure which language to use to answer the prompts, English (yes/no) or Bahasa Melayu (ya/tidak).

Criteria:

Participants verbalised their uncertainty of using either English or Bahasa Melayu.

Evidence:

Participant P6 was prompted on whether to have the cells right-justified. The participant was uncertain whether to use English or Bahasa Melayu.

[P6:E57-58] The system is asking *adakah anda ingin semua sel diselaraskan ke kanan?* ("Do you want the cells to be justified to the right?") I don't know what to answer, whether it's yes, no, Y, N or *tidak* ("no") *ya* ("yes") T and Y.

48. Mistook English as default instead of Bahasa Melayu

Description:

Participants were quite familiar with English software. Thus, participants pressed Y for Yes when it should be Ya *Yes,* N for No when it should be Tidak *No.*

Criteria:

Participants typing "Yes" or verbalising yes to answer prompts, that is, the participants were used to answering in English.

Evidence:

Participant P12 typed "Yes" at the prompt on whether to right-justify the cell.

[P12:E62] Adakan < > diselaraskan ke kanan? ("Do you want the cell to be justified to the right?") Ke kanan ("To the right?") umm.. Yes or No question? Oh that one. Don't know?

[P12:L] /-Format-b2-_CR_-b4-_CR_-yes-y-n-t

Participant P9 was asked to quit and not save the file. The participant used "n" for no instead of "t" for Tidak *No.*

[P9:81] Keluar ("Quit")... {Dan tidak perlu... ("no need to save")...} No... N... ESC...

[P9:L] /-Keluar-n-_CR_-_END_-_ESC_

49. Looking for Help

Description:

Help is not available in *Hamparan*. However, some participants attempted to look for help.

Criteria:

When using *Hamparan*, the participants typed F1, or typed "help" or verbalised they were looking for help or assistance.

Evidence:

From log files, participant P2 actually typed the word "help" into the cell.

[P2:E29] I'm thinking of aaa getting help from the software. Where should I go? ...

[P15:E24] Ummm.... okay, as the best way... to look for the formula... [Pause] Then find... then have to get help from the F one key, the help is not provided. Okay, then I press escape to go back to back to editing mode and try the F one again. Also, no. Okay, that means the help is not included. [Pause]

50. Error beep too loud

Description:

When a mistake is made, an error "Beep" is sounded. The "beep" was loud enough to startle a participant.

Criteria:

The beep was loud enough to make participant P4 "jump" in fright.

Evidence:

Participant P4 was observed to "jump" in fright when the error beep sounded after she entered a wrong value to a format prompt.

51. Looked for mouse

Description:

As participants were familiar with Window applications, they also expected to use the mouse in *Hamparan*. *Hamparan* does not support mouse use, and most participants looked to see if the mouse could be used.

Criteria:

Participants verbalised that they were looking to see if the mouse worked. Some participants were observed to move the mouse to see if the mouse could be used.

Evidence:

Participant P17 was observed to try the mouse three times. While completing the save task, he also commented that he was used to using the mouse:

[P17:E22] Arrr... I can't remember because I used to to... use this arr... mouse

At the start of the experiment, participant P4 commented:

[P3:E1] Mouse not working, is it? Urrrghh... [feigned disgust]

52. Looked for F10

Description:

Some participants, who are quite familiar with the spreadsheet, looked for F10 (equivalent to pressing ALT key in Windows applications to get at the menus).

Criteria:

Participant verbalised he was looking for F10.

Evidence:

Participant P17 was attempting the save task. He was looking for commands and menu to save the file. Thus, he looked for the F10 key.

[P17:E22] 'Kay F ten No

[P17:L] /-_F10_-_ESC_-_F10_-_ESC_

53. Looked for Undo

Description:

Most applications including Excel have Undo command. As such, the participants also looked for the undo facility. However, *Hamparan* does not have an undo function.

Criteria:

Participants verbalised they were looking for undo function.

Evidence:

Participant P8 wanted to delete a row but he accidentally deleted the column. He looked for the undo command.

[P8:E67] *Tak ada* undo [Laughs] ("Don't have undo") *Tak ada undo* ("Don't have undo")

54. Looked for AutoSum

Description:

Participants were looking for the AutoSum function (available in Excel) that automatically creates a formula to total up values in a column. *Hamparan* does not have this function.

Criteria:

Participants verbalised they were looking for this function.

Evidence:

While attempting to complete the formula task, participants P6 and P15 looked for the autosum feature.

[P6:E75] There is no option for copying or, ahh, automatic procedure... to do autosum or things like that.

[P15:E34] That means the system does not provide this apos... ah, selection features... the autosum features like Excel.

55. Looked for a way to select/highlight cells

Description:

The participants wanted to select or highlight a range of cells to change the cells' format or to obtain the sum of those cells.

Criteria:

Participants verbalised their intent to select/highlight the cells.

Evidence:

Participant P5 was trying to complete the currency format by highlighting:

[P5:E92] *Tapi* ("But")... you have to highlight it, isn't it? *Tapi tak boleh highlight 'ni* ("But cannot highlight this")... ... *Macam mana nak highlight semua?* ("How to highlight all this?")

56. Looked for Copy command

Description:

Some participants looked for the copy command in *Hamparan* to copy formulae instead of re-typing the formulae. *Hamparan* does not have the copy function.

Criteria:

Participants verbalised they were looking for the copy command.

Evidence:

Participants P1 and P3 were looking for the copy command to copy formulae to other cells. They were unable to find the copy command.

[P1:E75] Now, I would like to copy... First range, we try... to copy.

[P3:E35] Right, what I want to do is I want to copy from this place and paste it in the next one. ... That's not what I want. I want to try and copy...

57. Looked for filename on spreadsheet

Description:

When a spreadsheet is saved in Excel, the filename of the spreadsheet that has been saved will appear on the window. Participants tried to locate this filename to confirm that their file was saved successfully.

Criteria:

Participants verbalised that they were looking for the filename after they had just saved the spreadsheet.

Evidence:

After saving the file, participant P8 commented:

[P8:E33] Spreadsheet, already... already [Short laugh] {You're looking for what?} [Laughs] Looking for the name of the file [Laughs] I don't see it <wherever>. [Short laugh.] Very difficult to... to see the file. [Expect the name of the file to appear.]

Likewise, participant P15 was also looking for the filename.

[P15:E62] Umm.... the system does not showing... any... name for the file that I saved. I wonder whether the file has been saved or not.

58. Could not see gridlines

Description:

Participants expected to see the gridlines of the spreadsheet. *Hamparan* does not show the gridlines. The cursor appears as a rectangle on the screen showing the cell the cursor is on.

Criteria:

Participants commented that they could not see the gridlines.

Evidence:

Participant P11 was attempting to alter the column width but she commented that she could not see the "boxes" [gridlines].

[P11:E21] *Ruang kecil, ruang kecil.* ("Small space... small space") *Oh <> agaknya, saya ingin membesarkan tapi saya tak nampak itu <> ia* [Laughs] ("Oh... actually, I want to widen but I can't see... <> it [the gridlines]") *Saya tak nampak mana itu petak dia.* ("I can't see where the "boxes" [gridlines] are.")

59. Looked for Windows

Description:

Most software today have graphical user interfaces. Participants commented on the lack of windows.

Criteria:

Participants commented of the lack and the need for GUI.

Evidence:

[P2:E7] This is not a window version, I cannot see the icon.

60. Typed "Ya" or "Tidak" in full instead of just Y or T

Description:

At some prompts, users are supposed to type in Y for Ya *Yes* or T for Tidak *No.* However, the participants typed Ya or Tidak in full as there was no indication of what was the appropriate input.

Criteria:

The participant typed in the affirmative or negative answers in full as opposed to just the first character of the answer.

Evidence:

Participant P16 typed "TIDAK" *No* in full to answer the prompt. [after the T has been typed, the prompt would immediately dissapear. The participant would know that only one character was enough]

[P16:E44] *Adakah anda inginkan semua sel diselaraskan ke kanan?* ("Do you want all the cells to be justified to the right?") Oh this is margin... right? No... no... *tidak!* ("No!")

[P16:L] /-format-b5-_CR_-d5-_CR_-TIDAK-_CR_-Y-T

Participant P12 typed "yes" in full.

[P12:E62] *Adakan <mumble > diselaraskan ke kanan?* ("Do you want the cell to be justified to the right?") *Ke kanan* ("To the right?") Umm.. Yes or No question? Oh that one. Don't know?

[P12:L] /-format-b2-_CR_b4-_CR_-yes-y-nt

61. Looked for Ctrl-F

Description:

Participants looked for Ctrl-F key combination which participants hoped would show the available menus.

Criteria:

Participants verbalised they were looking for this key combination, or the participant used this key combination (as detected from the log files).

Evidence:

Participant P13 wanted to exit from the formatting prompt. She suggested using Ctrl-F probably because she was thinking about Exit command in the File menu.

[P13:E26] {If you want to... exit you want to cancel something, you have to... what key do you...?} Control-F?

62. Looked for Ctrl-S

Description:

Ctrl-S is the shortcut key combination to save the spreadsheet in spreadsheets such as Excel. Participants attempted to save the file using this key combination which does not work in *Hamparan.*

Criteria:

Participants verbalised they were looking for this key combination when they were completing the save the spreadsheet task.

Evidence:

Participants typed in various key combinations while trying to save the spreadsheet, one of these combinations was CTRL-S.

[P9:E30] ... Control... [Typing.] <Okay> {[Clear throat]} Oh, < > [Typing.] <Enter> [Laughs] Cannot...

[P9:L] Tried the following combinations Shift-F1, Shift-F5, Shift-F3, Shift-F1, Shift-F2, Ctrl-Q, Alt-Q, Ctrl-F1, Ctrl-N, F1, F2, F3, F4, S, Shift-F4, Ctrl-Shift S, Ctrl-S, W, Q, q, Ctrl-End

63. Looked for Ctrl-Q

Description:

The Ctrl-Q combination is another key combination which quits or exits from the more widely used software applications. Participants attempted to use this key combination to try and quit the spreadsheet program.

Criteria:

Participants typed the key combination.

Evidence:

In episode [P9:E30], while participant P9 was attempting to save the spreadsheet, she also tried CTRL-Q, ALT-Q. The participant was probably using this combination to quit (which would prompt users whether to save the file). This assumption to quit from the spreadsheet is supported in episode [P9:E34] whereby the participant found the spreadsheet and used the Keluar *Quit* command to quit and save the spreadsheet.

[P9:E30] ... Control... [Typing.] <Okay> {[Clear throat]} Oh, < > [Typing.] <Enter> [Laughs] Cannot...

[P9:L] Tried the following combinations Shift-F1, Shift-F5, Shift-F3, Shift-F1, Shift-F2, **Ctrl-Q, Alt-Q**, Ctrl-F1, Ctrl-N, F1, F2, F3, F4, S, Shift-F4, Ctrl-Shift S, Ctrl-S, W, Q, q, Ctrl-End

[P9:E34] ... And then *tekan* ("Press")... *keluar* ("Quit")... K... *Simpan hamparan ini?* ("Save this spreadsheet?")... Ya ("Yes.") [To the prompt]... .

[P9:L} /- _DOWN_-_RT_-_RT_-Keluar *Exit* -Ya-name\

Appendix I Usability Problems Isolated from Interview Data

1. Slow Performance

Participants found the spreadsheet to be "slow", that is, they had to type in formulae or type the slash key to access the commands.

[P1:Q1] Got to have slash before instruction appear on the screen. Not fast, have to go slash. Menu, [then] go to next [menu]. [interview notes]

[P5:Q1] ... And then this one, sometime you have to create your [own] formula <> *kan?* ("Right?") Takes a long time lah. Since Excel you just click with the formula, sum...

2. Help was not available

Participants suggested including some form of help to assist people use the spreadsheet. (The underlining has been added for emphasis.)

[P2:Q1] I didn't really have a good look at the whole screen before I start doing the work, I thought [I] was lost but after going into spreadsheet, then after going to spreadsheet and urr... I could know there are some help key to be used...

[P15:Q2] Okay, and if possible the system give a help to the system by following some... you can consider as quite standard convention like pressing the F one key. And maybe at least some help... if not a context sensitive help... they might have some help in order to help the user if they get lost. *[Laughs]* Okay

[P2:Q2] ... on the screen before you start doing any spreadsheet, urr, you can add it some more, add in some more ... urr guides or help, maybe in terms of icons or maybe simple letter that I think could help someone easily to start with the spreadsheet

[P16:Q2] Maybe can... and the helps command it should be quite details...

[P6:Q2] ... or there must be a help function somewhere... so that the user can can understand. Just, like just now, I am trying get to get through the formula... I don't know where to go [Laughs]

[P7:Q1] Umm... *sebab* ("because")... *kalau gunakan Excel ada tunjuk-tunjuklah* ("if use Excel there is pointers/help? ")... [Laughs]

3. Could not find menus

Participant suggested locating the commands/menu at the top of the screen like Excel. She suggested this since she probably could not find the menus where she expected them, that is, at the top of the screen.

[P5:Q2] *Atau kalau boleh... you just keluarkan perintah di atas 'tu... macam... macam... Excel lah. Keluarkan perintah dari sebelah atas... very clear... ... Why don't kita letakkan di atas?* ("Or if possible... you just show the commands at the top... like... like... Excel lah. Show the commands from the top... very clear... ... Why don't we place at the top [of the screen]?]")

4. Looked for mouse

Participants recommended the software should include the use of the mouse.

[P4:Q2] {Can you suggest any improvements to the software?} I think use mouse [laughs]

[P12:Q2] ... whatever, you know, maybe use mouse, I think much easier. Use mouse, mmm.

[P11:Q1] ... *biasa gunakan mouse tik tik tik ... sekarang cari mouse tak jumpa mouse* [Laughs] ... ("... accustomed to using a mouse tick tick tick [clicking noise of mouse]... now I look for mouse, I can't find the mouse. ...") [Laughs]

5. Looked for summation buttons (AutoSum)

Participant suggested a summation button to total up the the values.

[P13:Q1] ... whereby if like Lotus, you want to... umm... you want to... add all the total right? you need to press to press something like... ahh... a for sum and something like that... but this one is... it's easier but for someone who use to use Lotus it's difficult, you know? Okay?

6. Numerous functions were unavailable

Participants suggested including more spreadsheet functions such as AVERAGE and IF. The participants were unable to locate the functions they expected to find in the spreadsheet.

[P6:Q2] Like arr.., now what the market is going into, they have Lotus suite ahh, Lotus suites and Microsoft Office where from one software to a software, there is compatibility of data mmm that you can always cut and paste or import and export, so that it's flexible, and err...

[P12:Q2] ... More function, {*seperti*... ("like")} average, use if command, whatever, you know ...

[P15:Q1] ... there are quite a number of umm... what you call, the the features not following.. the the Lotus style and aaa why don't the system implement... aaa... all of the function under Lotus.

[P15:Q1] ... So really, there are some functions I never use and maybe I have

forgotten [Laughs] but the system provide that okay... especially maybe the function...

7. Commands were not clear

The participants commented that the commands were not clear and that the instructions/commands must be clear.

[P1:Q2] Simplify the words [commands]

[P3:Q2] ... I think improvements to the software, some of the words were not really clear what they mean.

[P8:Q2] ... It need to... umm, need to make an improvement hmm... what's hmm... aaa... very tsk [frustrated]...ahhh, ... *arahan-arahan yang lebih mudah lah.* ("commands that are much simpler")

[P9:Q2] ... *Macam* ("Like")... ahhh... *word yang senang sikit* ("words which are easier?") {*Maknanya word 'tu... makna word tu arahan kan?* ("By word you mean commands?")} *Arahan, ya ya arahan* ("Commands yes, yes, commands.") ...

[P17:Q2] In terms of Bahasa Melayu version just now... I think that the lettering or the command is quite ahh... confusing. ...

8. Bahasa Melayu terms problematic

Participants believed that they had problems with the spreadsheet because they lacked experience using Bahasa Melayu terminology. Both participant P3 and P6 conceded that when confronted with the Bahasa Melayu terms they depended on their knowledge of English software. As a result, participant P6 had to compare the Bahasa Melayu terms with English terms to understand those terms.

[P3:Q2] Apart from that, I think improvements to the software, some of the words were not really clear what they mean. Ah well, in most cases time what I was trying to do was to guess, guess what it could be based on my experience using English software. Even things like, You see the main thing is, I don't think in Malay [Bahasa Melayu] I guess, that's the reason why. ... ... Even the word *simpan* ("save") didn't really click on my mind that it is save, right? ...

[P6:Q2] Arr, okay. The language is good. The only problem is, it's my own vocabulary in in Bahasa Malaysia. ... Ahhh, every time I think of your terms, because of my experience, actually, I try to compare with the English terms what... what... what I use to.. to work with in Lotus. ...

9. Confusion between Yes/No or *Ya* ("Yes")/*Tidak* ("No")

Participants noted problems in answering the prompts, that is, whether to answer Yes or No answer, or *Ya* ("Yes") or *Tidak* ("No").

[P10:Q2] Umm... okay, the... *ya tidak* ("yes no")... ah... that's mean the question I asked, then there is no indication whether should put Yes or No. Because I think I am still tied up in the English version, so I put the No [Laughs] should be *Tidak* ("No.")

[P12:Q2] *Bila ada soalan ya kan? Buat* Yes or no*? Ya atau tidak?* ("When there is a question right? Put Yes or no? Ya ("Yes") or Tidak ("No")?") *Itu sajalah.* ("That's all.") *Kalau tidak, tak tahu kan?* ("If not, don't know right?") Only, ahhh, unless you're guessing that word, it's okay lah.

[P16:Q2] ... if you arr... want to answer yes or no... they should put, is it Y or slash T... {Ah... okay} [Laughs] Instead of guessing... [Laughs] whether is Y, T or yes no.

10. Unsure how to answer prompt, Y or Enter key

Participant noted that when prompted, she was unaware to use Yes or press the Enter key.

[P13:Q2] And the instruction like ummm... yes, ummm yes, do you want it to be *sebelah kanan* ("[justified] on the right side?") My answer is yes, but don't know which button to press whether it's enter or Y. {Okay} Something like that, make it clear.

11. Formula syntax unclear

Participant commented that a syntax for the formula probably because the syntax in FIRST was not clear.

[P15:Q2] I suspect that if I am not mistaken there is a format for the function... or maybe I use it in that format for the Lotus one two three [123], okay? Maybe in terms of two dots, they they are used for the range. The one I'm using the the Excel format... maybe the Lotus using the double double colon, or colon okay.

12. Lack of GUI

Participants suggested the use of GUI (Graphical User Interface). Participant P5 described the spreadsheet as "boring", unlike Excel (running on Windows) which "encourages" people to try the software.

[P5:Q2] *'Kay. Umm... Tapi. But... umm... ada perbezaanlah... bila* you *tengok Excel... <> Excel when you tengok Excel... tsk... you tak rasa boring tengok Excel...* [Laughs] *Bila* you *tengok hamparan 'tu,* you feel you *tak ada mood...* you know, you *tak ada mood nak... dia tak mengalakkan* you *untuk mencuba balik ataupun nak... nya...* you know... some, it creates something *yang boleh menarik orang... <> bagi perhatian sepenuhnya pada benda 'tu... Pada* that <> that *cell lah,* that *column...* that <> *hamparan 'tu.*

("But... umm... there are differences... when you see Excel... <> when you see Excel... tsk... you don't feel bored looking at Excel... [Laughs] When you see the spreadsheet, you feel you don't have mood... you know, you don't have mood to... the [spreadsheet's user interface] does not encourage you to try or to... you know... it creates something that can attract people's [attention]... <> complete attention to that cell, column, spreadsheet. ...")

Participant P11 believed that Windows (which includes the use of icons) would make the software easier to use.

[P11:Q2] *Umm...*, I think *banyak... sekarang, window-window* item... *yang lebih menyenangkan. ataupun bentuk gambar-gambar ka... macam sekarang 'ni ya... yang memudahkan* <> ("Umm..., I think many... today, items such as windows... are easier [to use]. ..... or in the form of pictures [icons]... like those that exists now... simpler...")

Participant P6 believed that FIRST should use a graphical user interface.

[P6:Q2] Ahh of course you have to <> using GUI and put other features. [Laughs] Put other features... arrr... like arr integrate with some other software.

13. No information about the column width unit

Participant identified that there was no information about the unit of the column width.

[P17:Q2] So we actually we don't know what's the actual... the actual *lebar* ("width") of the column.

14. Did not know how *Tunjukkan Rumus* Formula Display command worked

Participants was not aware of the Show Formula mode

[P6:Q2] When the system says *tunjukkan* ("show"), I cannot see the system toggle, until for quite a long time and then I know it's, it's actually toggling into the formula.

Participant P15 was not aware that "Formula Display" mode was on. When the mode is on, the formula is displayed in the cell and results of the calculation is displayed in the status bar. After the crash had returned the spreadsheet to default settings –mode off, when he saw the formula in the status bar and the results in the cell –what P15 terms as the norm.

[P15:Q1] Like for instance, the the display, of the results seems that first time appears at the status, I call that as a status bar, don't know what the name given to that... errr... which is the last last row on the screen. {umm hmm} And by right it should on the cell, the formula should appear at the bottom. But the second time after after the system

hang... when I go went in again... and the consis... the the it seems gives what I expect... [laughs] so [laughs] that's why this is what I mean by inconsistency... that's er very funny thing... [laughs]

15. Enter key did not work as expected

Participant pointed out that the Enter key did not work as he expected.

[P15:Q2] ... pressing the enter key normally should lead to the... to the following cell... or the cell below it ... Let's say by pressing enter key whether it should go left or right. Umm, I think most... this is my assumption, most people expect it go down but for this system, it just stay... just stay at current cell ...

16. Could not select cell range

Participant P15 noted that cells could not be selected and that suggested a method to input the range.

[P15:Q2] ... in terms of selecting the cell maybe the shift key followed by the down arrow, or left arrow, right arrow whatever right arrow, so that we can select the cell instead of keying in the... the range {Umm} and also the, the way the system asking for the range is quite funny, one by one. Why not we accept just the range by... by accepting, what I mean, by accepting the just one entry for the range... with one question, okay...

17. Message appeared too briefly

Participant commented that a message was displayed too briefly, before the user can read the message.

[P15:Q2] ... There are some improvements to be done because the message appear too fast... Urr... the user cannot get the message... I think this is even worse if you run the system on a faster machine [Laughs] So... the system should give some delay or just a pop up message and wait for the stu... wait for the user response urr... by pressing a key to confirm the message... so that the user know what happening to them.

18. Expected file name on Window

Participant expected to find the file name on the spreadsheet Window after he had saved the spreadsheet.

[P15:Q2] ... Umm, aa... another thing about this is a.... the system... aaa... should show the current file name. ... I'm not sure just now whether file has been saved or not until I really try with other option.

19. Menu navigation was not effective

Participant commented that the menu navigation was not effective. Unlike the conventional multi-level menus whereby, when Escape key is pressed, the spreadsheet is returned to the previous menu list, FIRST exits from the menu altogether.

[P15:Q2] I'm not sure how to go back to a previous menu after I go down to a level. So I have to press escape key based on my experience, escape normally can cancel. So I go back to the editing mode then I have to start all over again, by pressing back-slash to get to the menu again, and start all over again. The system I think should provide something like aaa... go back to a previous, previous menu.

20. Press / instruction was not clear

Participant commented that the instruction such as "Press / for list of commands" was not clear.

[P5:Q2] Umm.. instruction must be clear... ah... *memberi perintah 'tu hendaklah clear...* and then ah... *itulah... yang tadi tekan slash untuk perintah 'tu kan... kena* must be clear... so sometime *orang tak tahu... setiap maksud apanya kat situ...* what does it mean? ("Umm.. instruction must be clear... ah... giving instruction must be clear... and then ah... that... just now where we press slash for the commands right?... must be clear... so sometime "people" might not know... [they might ask] what's the meaning of that [command]... what does it mean?")

Appendix J Categories of Usability Problems

J.1 Categories of Think Aloud Usability Problems

The sections below categorise usability problems identified in think aloud data. For a detailed description of the usability problems, refer to Appendix H.

i. Inconsistencies with other Spreadsheet

Usability problems found under this category were related to the participants' experience of software applications they had used. The participants were looking for functions available in spreadsheets they had used before, such as Lotus 123 and Excel. For example, Usability Problem No. 53, the participants expected to find an Undo command. Undo command is available in some software applications, but not in FIRST.

2 Used arrow keys to select the command

5 Expected more sub-commands

7 Difficult to save the spreadsheet

33 Unable to locate *Keluar* ("Quit")

34 Need to press Enter or other arrow key to go to next cell

37 Navigation using tabs

38 Expected carriage return to move to the next cell

44 Treated DOWN and UP arrow key as CR key

51 Looked for mouse

52 Looked for F10

53 Looked for Undo

54 Looked for AutoSum

55 Looked for a way to select/highlight cells

56 Looked for Copy command

58 Could not see gridlines

59 Looked for windows

60 Looked for Ctrl-F

62 Looked for Ctrl-S

63 Looked for Ctrl-Q

ii. Unclear Commands

Participants had problems with the translated commands. In most cases, the participants did not know what the commands meant.

9 Did not know meaning of *Muatkan* ("Load")

13 Mistook *Hapus* Delete for deleting rows or columns

14 Did not know meaning of Goto

16 Did not know meaning of *Lajur* ("Column")

19 Did not know meaning of *Baris* ("Row")

20 Did not know meaning of *Sisipkan* ("Insert")

21 Confused between *Baris* ("Row") and *Lajur* ("Column")

22 Did not know meaning of *Sunting* ("Edit")

24 Did not know meaning of *Utiliti*

26 *Hitung Semula* ("Recalculate") command is not intuitive

28 Confused between /-Utiliti- *Tunjukkan Rumus* ("Formula Display and /-Utiliti- *Hitung Semula* ("Recalculate")

29 *Tunjukkan Rumus* ("Formula Display") command was not intuitive

30 Did not know meaning of Auto

iii. Spreadsheet Problems

The usability problems listed here pertain to program errors and the spreadsheet interface design per se. For example, Usability Problem No. 40, Unexpected Exit, is a program error whereby the keystrokes /-Left_Arrow is equivalent to a */-Keluar* ("Quit") instruction.

1 Slash / was not clear

35 Formula syntax was not intuitive

36 Increasing column width was not straightforward

40 Unexpected exit sequence

41 Crashed at Formatting

42 Unexpected crash at Column-Width

43 Unexpected load sequence /-hamparaN-_RT_

45 Could not delete

50 Error beep too loud

iv. Unclear Dialogue Messages/Prompts

The usability problems in this category pertain to prompts in the *Hamparan* spreadsheet. The participants did not understand/could not answer many of the prompts displayed.

10 Difficult to interpret prompt "Enter the first cell you wish to format:"

11 Difficult to answer prompt *"Adakah anda mahu semua sel diselaraskan ke kanan?"* ("Do you want the cell right-justified?")

12 Did not understand the prompt *"Berapa tempat perpuluhan yang perlu di bundarkan untuk setiap nombor?"* ("How many decimal places should the number be rounded to?")

15 Did not understand the Goto prompt

17 Unit of the cell width not known

18 Did not understand the prompt *Taipkan lebar lajur baru:* ("Enter the new column width:")

61 Typed *Ya* ("Yes") or *Tidak* ("No") in full instead of just "Y" or "T"

v. Poor Feedback

The list below comprises concerns with feedback after commands were selected. For example, Usability Problem No. 8: the message "Loading..." appeared too briefly to be read, after the load command was selected.

8 Message "Loading file ..." displayed too briefly

23 No feedback to *Sunting* ("Edit")

25 No feedback to *Hitung Semula* ("Recalculate")

27 *Tunjukkan Rumus* ("Formula Display") mode was not obvious

31 Feedback of Automatic Recalculation was not obvious

57 Looked for filename on Spreadsheet

vi. Language

The following problems pertain to language issues due to translation of the spreadsheet text messages to Bahasa Melayu. For example, Usability Problem No. 47: participants were confronted with prompts requiring a yes or no answer, the participants were unsure whether to enter a yes/no or a *Ya* ("Yes")/*Tidak* ("No") answer.

32 Thought *Keluar* was to open/Load (In Bahasa Melayu keluar also means "take/bring out")

46 Had to translate all the Bahasa Melayu terms to English

47 Not sure whether to answer yes no or ya ("Yes")/*Tidak* ("No")

48 Mistook English as default instead of Bahasa Melayu

vii. Problems with Modes

Editing and navigation mode, some participants had problems with the various modes. Example, Usability Problem No. 3, participants were unable to select commands because the participants were unaware that the spreadsheet was in editing mode.

3 Confused between command and editing mode

39 Confused between editing mode and navigation mode

viii. No Clear Exits

Some participants did not know how to exit from the prompt or the command mode, for example they did not use the "escape" key. Some participants thought the ESC key would crash/exit the spreadsheet and they would lose their work.

6 Did not know how to exit from prompt/command mode

ix. No Help

Some participants searched for a help command to assist them in completing the tasks.

49 Looking for Help

x. Problem with Shortcuts

Due to translation, shortcut keys "N" and "H" were used to allow the selection of the *hamparaN* ("Spreadsheet") and *Hapus* ("Delete") command instead of "S" and "D" for Spreadsheet and Delete. Participants pressed "H", to select *hamparaN* ("Spreadsheet") the first letter of the Bahasa Melayu command instead of "N".

4 Mix-up between hamparaN and Hapus command

J.2 Categories of Interview Usability Problems

Categories	UP #	Interview: Usability Problems
Inconsistencies with other Spreadsheet	1.	Slow Performance
	3.	Could not find menus
	4.	Looked for mouse
	5.	Looked for summation buttons (AutoSum)
	6.	Numerous functions were unavailable
	11.	Formula syntax unclear
	12.	Lack of GUI
	15.	Enter key did not work as expected
	16.	Could not select cell range
	19.	Menu navigation was not effective
Unclear Commands	7.	Commands were not clear
Spreadsheet Problems	20	Press / instruction was not clear
Unclear Dialogue Messages/Prompts	10.	Unsure how to answer prompt, Y or enter
	13.	No information about the column width unit
Poor Feedback	17.	Message appeared too briefly
	18.	Expected file name on window
Language Problems	8.	Bahasa Melayu terms problematic
	9.	Confusion between yes/no and ya/tidak
Mode Problems	14.	Did not know how *Tunjukkan Rumus* Formula Display command worked
No Clear Exits		
No Help	2.	Help was not available
Shortcut Problems		

UP# is interview usability problem number found in Appendix I.

Appendix K Spearman's Ranked Correlation

Spearman's Ranked Correlation Coefficient (and the respective p-value – in brackets) of all participants and the respective sub-groups (n = number of participants in the sub-group)

	UAT Combination pairs		
	TA-SUS	**TA-Intw**	**SUS-Interview**
a. All participants (n=17)	0.473 (0.055)	0.349 (0.170)	0.378 (0.135)
b. Lotus 123 (DOS) Experience			
No experience (n=8)	0.168 (0.691)	0.048 (0.911)	0.012 (0.978)
With experience * (n=9)	0.784 (0.012)	0.899 (0.001)	0.695 (0.038)
c. Familiar with Experimenter			
No (n=9)	-0.036 (0.932)	-0.071 (0.867)	0.205 (0.627)
Yes * (n=8)	0.778 (0.014)	0.850 (0.04)	0.703 (0.035)
d. DOS			
No (n=6)	-0.551 (0.257)	0.086 (0.872)	0.493 (0.321)
Yes (n=11)	0.851 (0.001)	0.493 (0.123)	0.381 (0.248)
e. Years Used			
Less than 5 (n=7)	0.074 (0.875)	0.393 (0.383)	-0.111 (0.812)
More than 9 (n=10)	0.579 (0.079)	0.900 (0.000)	0.443 (0.199)
f. Total Hours of Spreadsheet-use			
≤ 40 hours (n=4)	-0.800 (0.200)	-0.400 (0.600)	0.200 (0.800)
More than 40 hours (n=13)	0.473 (0.055)	0.349 (0.170)	0.378 (0.135)
g. Status			
Low (n=7)	0.670 (0.100)	0.107 (0.819)	0.059 (0.900)
High (n=10)	0.476 (0.165)	0.588 (0.074)	0.543 (0.105)
h. Gender			
Female (n=9)	0.366 (0.333)	0.494 (0.177)	0.712 (0.031)
Male (n=8)	0.596 (0.119)	0.133 (0.754)	-0.127 (0.765)
i. Age			
≤ 26 years old (n=8)	0.814 (0.014)	0.095 (0.823)	0.287 (0.490)
Between 27 and 36 years old (n=9)	0.097 (0.804)	0.025 (0.949)	0.599 (0.088)

* All correlations were statistically significant at 5% significance level

References

Ackerman, M., Starr, B., Hindus, D. and Mainwaring, S. (1997). Hanging on the Wire: A Field Study of An Audio-Only Media Space. *ACM Transactions on Computer-Human Interaction.* Vol. 4. No. 1. p39-66.

Apple Computer. (1992). *Guide to Macintosh Software Localization.* Reading, Massachusetts, USA: Addison Wesley.

Asma Abdullah. (1996). *Going Glocal: Cultural Dimensions in Malaysian Management.* Shah Alam, Malaysia: Malaysian Institute of Management, SNP Offset.

Barbour, R. H. (1996). Te Ripanga = the Spreadsheet. *Proceedings of the 6th Biennial Conference of the New Zealand Computers in Education Society.* (Hamilton, New Zealand, 14-17 January). p66-76.

Barbour, R. H. and Yeo, A. (1996). Internationalising a Spreadsheet for the Pacific Basin Languages. *New Zealand Journal of Computing.* Vol. 6. No. 1.

Bias, R. and Mayhew, D. (Eds.). (1994). *Cost-justifying Usability.* Boston, Massachusetts, USA: Academic Press.

Bødker K, Pedersen J S. (1991). Workplace Cultures: Looking at Artifacts, Symbols and Practices. In Greenbaum J, Kyng M (Eds.). *Design at Work: Cooperative Design of Computer Systems.* Hillsdale, New Jersey, USA: Lawrence Erlbaum Associates. p121-136.

Bourges-Waldegg, P. and Scrivener, S. (1998). Meaning, the Central Issue in Cross-Cultural HCI Design. *Interacting with Computers.* Vol. 9. No. 3. p287-310.

Bourges-Waldegg, P., Moreno, L. and Rojano, T. (1999). Internationalization and Localization of Educational Software. In Prabhu, G. and Del Galdo, E. (Eds.). *Designing for Global Markets.* Proceedings of First International Workshop on Internationalization of Products and Systems (IWIPS-99). (Rochester, New York, USA, 20-22 May). Rochester, New York, USA: Backhouse Press.

Brooke, J. (1996). SUS: A Quick and Dirty Usability Scale. In Jordan, P., Thomas, B., Weerdmeester, B. and McClelland, I. (Eds.). *Usability Evaluation in Industry.* Taylor and Francis. p189-194.

Brooke, J. (1998). Email, to Usability Test Mailing List, commenting on relationship of SUS and usability.

Burns, R. (1994). *Introduction to Research Methods.* Second Edition. Melbourne, Australia: Longman Cheshire.

Carey, J. (1998). Creating Global Software: A Conspectus and Review. *Interacting with Computers.* Vol. 9. No. 4. p449-466.

Checkland, P. (1981). *Systems Thinking, Systems Practice.* Chichester, UK: John Wiley and Sons.

Conger, S. (1994). *The New Software Engineering.* Belmont, California, USA: International Thomson Publishing.

Davison, R., Vogel, D., Harris, R. and Jones, N. (2000). *Electronic Journal on Information Systems in Developing Countries.* Vol. 1. Available as http://www.is.cityu.edu.hk/Research/ejisdc/. Last sighted on 14 November 2001.

Day, D. (1998b). Shared Values and Shared Interfaces: The Role of Culture in the Globalisation of Human-Computer Systems–Editorial. *Interacting with Computers.* Vol. 9. No. 3. p269-274.

Day, D. (1999). Thinking Differently, Acting Together: A Treatise on Technology Acceptance in the Era of Internationalization. In Prabhu, G. and Del Galdo, E. (Eds.). *Designing for Global Markets.* Proceedings of First International Workshop on Internationalization of Products and Systems (IWIPS-99). (Rochester, New York, USA, 20-22 May). Rochester, New York, USA: Backhouse Press. p25-43.

Day, D. (Ed.). (1998a). Shared Values and Shared Interfaces: The Role of Culture in the Globalisation of Human-Computer Systems. *Interacting with Computers.* Vol. 9. No. 3.

Day, D. (Ed.). (1998c). Shared Values and Shared Interfaces: The Role of Culture in the Globalisation of Human-Computer Systems. *Interacting with Computers.* Vol. 9. No. 4.

Del Galdo, E. (1990). Internationalization and Translation: Some Guidelines for the Design on Human Computer Interfaces. In Nielsen, J. (Ed.). *Designing User Interfaces for International Use.* New York, USA: Elsevier. p1-10.

Del Galdo, E. and Nielsen, J. (Eds.). (1996). *International User Interfaces.* New York, USA: John Wiley and Sons.

Dewan Bahasa dan Pustaka. (1993). *Istilah Komputer.* Kuala Lumpur, Malaysia: Dewan Bahasa dan Pustaka.

Dix, A., Finlay, J., Abowd, G. and Beale, R. (1998). *Human-Computer Interaction.* Englewood Cliffs, New Jersey, USA: Prentice Hall.

Donahue, G. (2001). Usability and the Bottom Line. *IEEE Software.* Vol. 18. No. 1. p31-37.

Doubleday, A., Ryan, M., Springett, M. and Sutcliffe, A. (1997). A Comparison of Usability Techniques for Evaluating Design. *Proceedings of the Conference on Designing Interactive Systems: Process, Practices, Methods, and Techniques.* ACM Press. p101-110.

Dray, S. (1995). The Importance of Designing Usable Systems. *interactions.* Vol. 2. No. 1. p17-20.

Dray, S. (1996). Designing for the Rest of the World: A Consultant's Observation. *interactions.* Vol. 3. No. 2. p15-18.

Dray, S. and Mrazek, D. (1996). A Day in the Life: Studying Context Across Cultures. In Del Galdo, E. and Nielsen, J. (Eds.). *International User Interfaces.* New York, USA: John Wiley and Sons. p242-256.

Dumas, J. and Redish, J. (1993). *A Practical Guide to Usability Testing.* New Jersey, USA: Ablex Publishing Corporation.

Ethnologue. (2001). *Ethnologue: Languages of the World.* Fourteenth Edition. Available as http://www.ethnologue.org. Last sighted on 14 November 2001.

Evers, V. & Day, D. (1997). The Role of Culture in Interface Acceptance. In Howard, S., Hammond, J. & Lindgaard, G. (Eds.). *Proceedings of the 6th IFIP TC13 International Conference on Human-Computer Interaction.* (Sydney, Australia, 14-18 July). p260-267.

FAO/World Bank. (2001). *Botswana: Population.* Available as http://www.fao.org/WAICENT/faoinfo/economic/giews/english/basedocs/bot/botpop1e.stm. Last sighted on 11 November 2001.

Fenton, N., Pfleeger, S. and Glass, R. (1994). Science and Substance: A Challenge to Software Engineers. *IEEE Software.* Vol. 11. No. 4. p86-95.

Fernandes, T. (1995). *Global Interface Design.* Chestnut Hill, USA: AP Professional.

Gaudart, H. (1995). Some Ways of Speaking Among Mother-tongue Speakers of English in Malaysia. In Zainab A. M., Z. and Baskaran, L. M. (Eds.). *Verbal Interactions at Play: Rules of Speaking.* Petaling Jaya, Malaysia: Pelanduk Publications. p26-46.

Glass, R. (1994). The Software-Research Crisis. *IEEE Software.* Vol. 11. No. 6. p42-47.

Griffiths, D., Heppell, S., Millwood, R. and Mladenova, G. (1994). Translating Software: What It Means and What It Costs for Small Cultures and Large Cultures. *Computers in Education.* Vol. 22. No. 1/2. p9-17.

Guyatt, G., Walter, S., Shannon, H., Cook, D., Jaeschke, R. and Heddle, N. (1995). Basic Statistics for Clinicians: Correlation and Regression. *Canadian Medical Association Journal.* Vol. 152. No. 4. p497-504. Available as http://www.cma.ca/cmaj/vol-152/issue-4/0497.htm. Last sighted on 15 November 2001.

Hall, A. (1962). *A Methodology for Systems Engineering.* Princeton, New Jersey, USA: Van Nostrand Reinhold.

Hall, P. and Hudson, R. (Eds.). (1997). *Software Without Frontiers: A Multi-platform, Multi-cultural, Multi-nation Approach.* Chichester, UK: John Wiley and Sons.

Heiman, G. (1992). *Basic Statistics for the Behavioural Sciences.* Boston, Massachusetts, USA: Houghton Mifflin Co.

Henderson, R., Smith, M., Podd, J. and Varela-Alvarez, H. (1995). A Comparison of the Four Prominent User-based Methods for Evaluating the Usability of Computer Software. *Ergonomics.* Vol. 28. No. 10. p2030-2044.

Herman, L. (1996). Towards Effective Usability Evaluation in Asia: Cross-cultural Differences. In Grundy, J. and Apperley, M. (Eds.). *Proceedings of OZCHI'96: Sixth Australasian Computer Human Interaction Conference.* (Hamilton, New Zealand, 24-27 November). IEEE. p135-136.

Heylighen, F. (1998). Basic Concepts of the Systems Approach. *Principia Cybernetica Web.* Available as http://pespmc1.vub.ac.be/SYSAPPR.html. Last sighted on 14 July 2001.

Heylighen, F. and Joslyn, C. (1992). What is Systems Theory?. *Principia Cybernetica Web.* Available as http://pespmc1.vub.ac.be/SYSTHEOR.html. Last sighted on 14 July 2001.

Heylighen, F., Josyln, C. and Turchin, V. (1999). What are Cybernetics and Systems Science?. *Principia Cybernetica Web.* Available as http://pespmc1.vub.ac.be/CYBSWHAT.html. Last sighted on 14 July 2001.

Hix, D. and Hartson, R. (1993). *Developing User Interfaces: Ensuring Usability Through Product and Process.* New York, USA: John Wiley and Sons.

Hofstede, G. (1994). *Cultures and Organizations: Software of the Mind.* Paperback Edition. London, UK: HarperCollins.

Hoft, N. (1996). Developing a Cultural Model. In Del Galdo, E. and Nielsen, J. (Eds.). *International User Interfaces.* New York, USA: John Wiley and Sons. p41-73.

Holleran, P. (1991). A Methodological Note on Pitfalls in Usability Testing. *Behaviour and Information Technology.* Vol. 20. No. 5. p345-357.

Holyer, A. (1994). Methods for Evaluating User Interfaces. *Cognitive Science Research Paper No. 301, School of Cognitive and Computing Sciences, University of Sussex.* Available as http://www.cogs.susx.ac.uk/cgi-bin/htmlcogsreps?csrp301. Last sighted on 14 November 2001.

Honkela, T., Lehtola, A., Kalliomäki, S., Suitiala, R., Hudson, R., Karkaletsis, V. and Vouros, G. (1997). A Recommended Globalization Method. In Hall, P. and Hudson, R. (Eds.). *Software Without Frontiers: A Multi-platform, Multi-cultural, Multi-nation Approach.* Chichester, UK: John Wiley and Sons. p33-49.

Honold, P. (1999). Learning How to Use a Cellular Phone: Comparison Between German and Chinese Users. *Technical Communication.* Vol. 46. No. 2. p196-205.

Honold, P. (2000). Culture and Context: An Empirical Study for the Development of a Framework for the Elicitation of Cultural Influence in Product Usage. *The International Journal of Human-Computer Interaction.* Vol. 12. No. 3 & 4. p327-345.

HR Focus. (1993). More Companies Are Drawing Strength from Diversity. *HR Focus.* Vol. 70. No. 6. p2.

Hudson, R., McHugh and Kalpakas, C. (1997). How Major Software Companies Approach Globalization. *Software Without Frontiers: A Multi-platform, Multi-cultural, Multi-nation Approach.* Chichester, UK: John Wiley and Sons. p15-32.

ISO/DIS 9241-11. (1998). Ergonomic Requirements for Office Work with Visual Display Terminals (VDTs)–Part 11: Guidance on Usability.

ISO/IEC 12207: 1995/Amd 1: 2002 *Information Technology : Software Life Cycle Processes.*

Ito, M. and Nakakoji, N. (1996). Impact of Culture on User Interface Design. In Del Galdo, E. and Nielsen, J. (Eds.). *International User Interfaces.* New York, USA: John Wiley and Sons. p105-126.

Jackson, M. (1991). *Systems Methodology for the Management Sciences.* New York, USA: Plenum Press.

Jacobsen, N., Hertzum, M. and John, B. (1998). The Evaluator Effect in Usability Tests. In Karat, C. and Lund, A. (Eds.). *Proceedings of CHI 98: Conference on Human Factors in Computing Systems (Summary).* (Los Angeles, 18-23 April). ACM Press. p253-254.

Jandt, F. (2001). *Intercultural Communication: An Introduction.* Third Edition. London, UK: Sage Publications.

Jeffries, R., Miller, J., Wharton, C. and Uyeda, K. (1991). User Interface Evaluation in the Real World: A Comparison of Four Techniques. *Proceedings of CHI 91: Conference on Human Factors in Computing Systems.* p119-124.

John, B. and Marks, S. (1997). Tracking the Effectiveness of Usability Evaluation Methods. *Behaviour and Information Technology.* Vol. 16. No. 4/5. p188-202.

Johnson, J., Roberts, T., Verplank, W., Smith, D., Irby, C., Beard, M. and Mackey, K. (1989). The Xerox "Star": A Retrospective. *IEEE Computer.* Vol. 22. No. 9. p11-26.

Kaner, C., Falk, J. and Nguyen, H. (1993). *Testing Computer Software.* New York, USA: Van Nostrand Reinhold.

Kano, N. (1995). *Developing International Software for Windows 95 and Windows NT.* Microsoft Press. Available as http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnintl/html/s24aa.asp. Last sighted on 30 November 2001.

Karat, C. (1994). A Business Case Approach to Usability Cost Justification. In Bias, R. and Mayhew, D. (Eds.). *Cost-justifying Usability.* Boston, Massachusetts, USA: Academic Press. p46-70.

Karat, C., Campbell, R. and Fiegel, T. (1992). Comparison of Empirical Testing and Walkthrough Methods in User Interface Evaluation. *Proceedings of CHI 92: Conference on Human Factors in Computing Systems.* (Monterey, California, USA, 3-7 May). ACM Press. p397-404.

Kennelly, C. (1991). Digital Equipment Corporation. *Digital Guide to Developing International Software.* Bedford, Massachusetts, USA: Digital Press.

Kissel, G. (1995). The Effect of Computer Experience on Subjective and Objective Software Usability Measures. (Eds.). *Proceedings of CHI 95: Conference on Human Factors in Computing Systems.* (Denver, Colorado, USA, 7-11 May). ACM Press. p284-285.

Kokkotos, S. and Spyropoulos, C. An Architecture for Internationalization. In Hall, P. and Hudson, R. (Eds.). *Software Without Frontiers: A Multi-platform, Multi-cultural, Multi-nation Approach.* Chichester, UK: John Wiley and Sons. p111-121.

Kurosu, M. (2001). Personal Communication.

Kurosu, M., Motoki, K. and Itoh, Y. (2001). Comparative Study of the Internationalized Software – Do They Really Fit to the Targeted User?. *Proceedings of HCI International 2001: Systems, Social and Internationalization Design Aspects of Human-Computer Interaction.* (New Orleans, USA, 5-10 August). p519-523.

Landauer, T. (1995). *The Trouble with Computers.* Cambridge, Massachusetts, USA: The MIT Press.

Lehtola, A., Kalliomäki, S., Honkela, T. and Lillqvist, T. (1997). An Application Framework for Internationalization. In Hall, P. and Hudson, R. (Eds.). *Software Without Frontiers: A Multi-platform, Multi-cultural, Multi-nation Approach.* Chichester, UK: John Wiley and Sons. p83-110.

Levin, J. and Fox, J. (1997). *Elementary Statistics in Social Research.* Seventh Edition. New York, USA: Addision Wesley.

Lim, K. Y. and Usma, M. (1998). Usability Evaluation in the Field: Lessons from a Case-Study Involving Public Information Kiosks. *Proceedings of Asia Pacific Computer Human Interaction Conference.* IEEE. p20-75.

Livermore, C. and Coronado, J. (2001). Growing Pains of Designing Global Products. Day, D. and Dunckley, L. M. (Eds.). *Proceedings of Third International Workshop on Internationalization of Products and Systems (IWIPS-01).* (Milton-Keynes, United Kingdom, 12-14 July). p47-60.

Luong, T., Lok, J. and Driscoll, K. (1995). *Internationalization: Developing for Global Markets.* Wiley and Sons Ltd.

Macro, A. (1990). *Software Engineering: Concepts and Management.* New York, USA: Prentice Hall.

Madell, T., Parsons, C. and Abegg, J. (1994). *Developing and Localizing International Software.* Englewood Cliffs, New Jersey, USA: Prentice Hall.

Marcus, A. (1993). Human Communications Issues in Advanced UIs. *Communications of the ACM.* Vol. 36. No. 4. p100–109.

McHugh, N., Honkela, T. and Hudson, R. (1997). Quality Assurance. In Hall, P. and Hudson, R. (Eds.). *Software Without Frontiers: A Multi-platform, Multi-cultural, Multi-nation Approach.* Chichester, UK: John Wiley and Sons. p219-228.

Merriam-Webster. (1997). *Merriam-Webster Online: The Language Centre. International Business Communication: Malaysia.* Available as http://www.m-w.com/info/buscorr/Address00000025.html. Last sighted on 19 November 2001.

Modica, C. and Fiedler, K. (1999). Tips for Successful International Evaluation of Localized User Interfaces. In Prabhu, G. and Del Galdo, E. (Eds.). *Designing for Global Markets.* Proceedings of First International Workshop on Internationalization of Products and Systems (IWIPS-99). (Rochester, New York, USA, 20-22 May). Rochester, New York, USA: Backhouse Press.

Myers, B. and Rosson, M. (1992). Survey on User Interface Programming. *Proceedings of the CHI 92: Conference on Human Factors in Computing Systems.* ACM Press. p195-202.

Mynatt, B. (1990). *Software Engineering with Student Project Guidance.* Englewood Cliffs, New Jersey, USA: Prentice Hall.

Nakakoji, K. (1993). Cross-cultural Considerations in Designing Human-Computer Interaction. *American Programmer.* Vol. 6. No. 10. p18-24.

Nakakoji, K. (1994). Crossing the Cultural Boundary. *BYTE.* Vol. 19. No. 6. p107-109.

Neuman,W.(1994). *Social Research Methods: Qualitative And Quantitative Approaches.* Needham Heights, Massachusetts, USA: Allyn & Bacon.

Nielsen J. (1990b). *Usability Testing of International Interfaces.* In Nielsen, J. (Ed.). *Designing User interfaces for International Use.* New York, USA: Elsevier. p39-44.

Nielsen, J. (1993). *Usability Engineering.* Paperback Edition. Boston, Massachusetts, USA: Academic Press.

Nielsen, J. (1996). International Usability Engineering. In Del Galdo, E. and Nielsen, J. (Eds.). *International User Interfaces.* New York, USA: John Wiley and Sons. p1-19.

Nielsen, J. (Ed.). (1990a). *Designing User Interfaces for International Use.* New York, USA: Elsevier.

Nielsen, J. and Levy, J. (1994). Measuring Usability: Preference vs. Performance. *Communications of the ACM.* Vol. 37. No. 4. p66-75.

O'Donnell, S. (1994). *Programming for the World: A Guide to Internationalization.* Englewood Cliffs, New Jersey, USA: Prentice Hall.

Onibere, E., Morgan, S., Busang, E. and Mpoeleng, D. (2001). Human–computer Interface Design Issues for A Multi-cultural and Multi-lingual English Speaking Country–Botswana. *Interacting with Computers.* Vol. 13. No. 4. p497-512.

Porteous, M., Kirakowski, J. and Corbett, M. (1993). *SUMI User Handbook.* Human Factors Research Group, University College Cork, Ireland.

Prabhu, G., Chen, B., Bubie, W. and Koch, C. (1997). Internationalization and Localization for Cultural Diversity. In Salvendy, G., Smith, M. and Koubek, R. (Eds.). *Advances in Human Factors Ergonomics, 21A. Design of Computing Systems: Cognitive Considerations.* Elsevier. p149-152.

Preece, J., Rogers, Y., Sharp, H., Benyon, D., Holland, S. and Carey, T. (1994). *Human-Computer Interaction.* Workingham, UK: Addison Wesley.

Pressman, R. (1997). *Software Engineering: A Practitioner's Approach.* Fourth Edition. New York, USA: McGraw-Hill.

Rai, S. (2001). Bridging the Gap. *Time Magazine.* Vol. 157. No. 22. Available as http://www.time.com/time/interactive/society/gap.html. Last sighted on 14 November 2001.

Robinson, H. (2001). Reflecting on Research and Practice. *IEEE Software.* Vol. 18. No. 1. p110–112.

Russo, P. and Boor, S. (1993). How Fluent is your Interface? Designing for International Users. *Proceedings of INTERCHI'93: Conference on Human Factors in Computing Systems.* (Amsterdam, Netherlands, 24-29 April). ACM Press. p342-347.

Sacher, H. (1998). Interactions in Chinese: Designing Interfaces for Asian Languages. *interactions.* Vol. 5. No. 5. p28-38.

Schermerhorn, J. and Bond, M. (1997). Cross-Cultural Leadership Dynamics in Collectivism and High Power Distance Settings. *Leadership and Organization Development Journal.* Vol. 18. p187-193.

Schmitt, D. (2000). *International Programming for Microsoft Windows.* Microsoft Press.

Shapiro, S. (1997). Splitting the Difference: The Historical Necessity of Synthesis in Software Engineering. *IEEE Annals of the History of Computing.* Vol. 19 No. 20. p20-54.

Shneiderman, B. (1992). *Designing the User Interface: Strategies for Effective Human-Computer Interaction.* Second Edition. Reading, Massachusetts, USA: Addison Wesley.

Sommerville, I. (1996). *Software Engineering.* Fifth Edition. Wokingham, UK: Addison Wesley.

Sukaviriya, P. and Moran, L. (1990). User Interfaces for Asia. In Nielsen, J. (Ed.). *Designing User Interfaces for International Use.* New York, USA: Elsevier. p189-218.

Sweeney, M. and Dillon, A. (1987). The Application of Psychological Techniques to the Evaluation of Human Computer Interaction. In Bullinger, H. and Shackel, B. (Eds.). *Proceedings of IFIP INTERACT'87: Human-Computer Interaction.* Amsterdam, Netherlands: North Holland. p367-373.

Taylor, D. (1990). Creating International Applications: A Hands-On Approach Using the Hewlett-Packard NLS Package. In Nielsen, J. (Ed.). *Designing User Interfaces for International Use.* New York, USA: Elsevier. p123-158.

Taylor, D. (1992). *Global Software: Developing Applications for the International Market.* New York, USA: Springer-Verlag.

Teasley, B., Leventhal, L., Blumenthal, B., Instone, K. and Stone, D. (1994). Cultural Diversity in User Interface Design: Are Intuitions Enough?. *SIGCHI Bulletin.* Vol. 26. No. 1. p36-40.

Thanasankit, T. and Corbitt, B. (2000). Cultural Context and Its Impact on Requirements Elicitation in Thailand. *Electronic Journal on Information Systems in Developing Countries.* Vol. 1. Available as http://www.is.cityu.edu.hk/Research/ejisdc/vol1.htm. Last sighted on 15 November 2001.

The Star. (1998). Quotes of the Week: 20 July 1998. *The Star Online.* Available as http://www.thestar.com.my/. Last sighted 20 July 1998.

Trollip, S. and Brown, G. (1987). Designing Software for Easy Translation into Other Languages. *Journal of Computer-Based Instruction.* Vol. 14. No. 3. p119-123.

Tuthill, B. (1993). *Solaris International Developer's Guide.* Englewood Cliffs, USA: Prentice Hall.

UNHDP. (2001). *United Nations Human Development Report 2001: Making New Technologies Work for Human Development.* Available as http://www.undp.org/hdr2001/. Last sighted on 14 November 2001.

Uren, E. (1998). Annotated Bibliography on Internationalization and Localization. *The Journal of Computer Documentation.* Vol. 21. No. 1. p26-33.

Uren, E., Howard, R. and Perinotti, T. (1993). *Software Internationalization and Localization: An Introduction.* New York, USA: Van Nostrand Reinhold.

Valacich, J., George, J. and Hoffer, J. (2001). *Essentials of Systems Analysis and Design.* Upper Saddle River, New Jersey, USA: Prentice Hall.

Van Gigch, J. (1974). *Applied General Systems Theory.* New York, USA: Harper and Row.

Van Vliet, H. (1993). *Software Engineering: Principles and Practice.* Chichester, UK: John Wiley and Sons.

Von Bertalanffy, L. (1968). *General Systems Theory: Foundations, Development, Applications.* New York, USA: George Braziller.

Wright, P. and Monk, A. (1991). A Cost-Effective Evaluation Method for Use by Designers. *International Journal of Man-machine Studies.* Vol. 35. No. 6. p891-912.

X/Open. (1993). *Open Guide: Internationalisation Guide, Version 2.* Reading Berkshire, UK: X/Open Co.

Yamagishi, N. and Azuma, M. (1987). Experiments on Human-computer Interaction Evaluation. In Salvendy, G. (Ed.). *Cognitive Engineering in the Design of Human Computer Interaction.* Elsevier Science. p167-174.

Yeo, A. (1998). Cultural Effects In Usability Assessment. In Karat, C. and Lund, A. (Eds.). *Proceedings of CHI 98: Conference on Human Factors in Computing Systems (Summary) – Doctoral Consortium.* (Los Angeles, April). ACM Press. p74-75.

Yeo, A. (2001). Global-Software Development Lifecycle: An Exploratory Study. In Jacko, J., Sears, A. Beaudouin-Lafon, M. and Jacob, R. (Eds.). *Proceedings of CHI 01: Conference on Human Factors in Computing Systems.* (Seattle, USA, 31 March-5 April). ACM Press. p104-111.

Yeo, A. and Barbour, R. H. (1997). Localising a Spreadsheet: An Iban Example. *Malaysian Journal of Computer Science.* Vol. 10. No. 2. p38-44.

Yeo, A., Barbour, R. H. and Apperley, M. (1998). Cultural Influence In Usability Assessment. In Hanson, M. (Ed.). *Contemporary Ergonomics 1998.* (Cirencester, Gloustershire, UK, April). Taylor and Francis. p274-278.

Yong, S. H. (1995). *Siri Literasi Komputer Fajar Bakti: Lotus 1-2-3.* Shah Alam, Malaysia: Penerbit Fajar Bakti Sdn. Bhd.

Zabid A. R. M., Anantharaman, R. N. and Raveendran, J. (1997). Corporate Cultures and Work Values in Dominant Ethnic Organisations in Malaysia. *Journal of Transnational Management Development.* Vol. 2. No. 4. p51-63.