



BLEKINGE INSTITUTE OF TECHNOLOGY

Department of Computer Science and Software Engineering

DVC001 - Bachelor Thesis, Spring 2002

Global Software Development



By

Anatoly Chervyakov
Andreas Hellström

Examinator and supervisor(s):
Guohua Bai
Laurence Henesey

ABSTRACT

Global Software Development is an area that has been recently highlighted. Translation is usually an issue that people associate with adapting software for a specific country but it is much more than that. Through text representation and data formatting you can go deep into cultural aspects that affect the user interface and even the core functionality.

In this thesis we will discuss just how these matters can affect development of global software. To carry out the investigation we analyzed current approaches and references and build up an extended methodology with suggestions to develop global software applications.



ACKNOWLEDGEMENT

First of all would like to thank our examiner and supervisor Gouhua Bai and additionally Laurence Henesey for their help, patience and guidance.

We would like as well express our thanks to Carrie Livermore and Jose Coronado at Hyperion, who made a great contribution to the thesis. Their invaluable assistance made our work easier and inspired us during the two long month of writing. Beside that we would like to thank Alvin M. Yeo (University of Waikato, New Zealand).

TABLE OF CONTENTS

1. INTRODUCTION.....	5
1.1. PURPOSE	6
1.2. GOAL	6
1.3. PROBLEM	6
1.4. RESEARCH QUESTIONS	7
1.5. INPUT	7
1.6. OUTPUT	8
1.7. TARGET AUDIENCE	8
1.8. DELIMITATION.....	8
1.9. METHOD.....	9
2. BACKGROUND	10
2.1. THE IMPORTANCE OF GLOBALIZATION.....	10
2.2. ASPECTS OF GLOBALIZATION.....	11
2.2.1. <i>Text representation</i>	11
2.2.2. <i>Data formatting</i>	12
2.2.3. <i>Cultural aspects</i>	12
3. HOFSTEDE'S CULTURAL DIMENSIONS.....	15
3.1. HOFSTEDE'S DIMENSIONS.....	15
3.1.1. <i>Power distance</i>	15
3.1.2. <i>Individualism or collectivism</i>	17
3.1.3. <i>Uncertainty Avoidance</i>	18
3.1.4. <i>Masculinity-Femininity</i>	18
3.1.5. <i>The long or short term orientation (LT)</i>	19
4. ESTABLISHED APPROACHES	20
4.1. ORGANIZATIONAL ISSUES	21
4.1.1 <i>Carey's approach</i>	21
4.1.2. <i>Hyperion solution</i>	21
4.1.3 <i>Yeo's approach</i>	23
4.2. CULTURAL ISSUES	23
4.2.1 <i>Carey's approach</i>	23
4.2.2 <i>Hyperion solution</i>	23
4.2.3. <i>Yeo's approach</i>	24
4.3. INTERNATIONAL INTERFACE	24
4.3.1 <i>Carey's approach</i>	24
4.3.2 <i>Hyperion solution</i>	25
4.3.3. <i>Yeo's approach</i>	25
4.4. DOCUMENTATION	25
4.4.1. <i>Carey's approach</i>	25
4.4.2. <i>Hyperion solution</i>	26
4.4.3 <i>Yeo's approach</i>	26
4.5. TESTING, QA	26
4.5.1 <i>Carey's approach</i>	26
4.5.2. <i>Hyperion solution</i>	27
4.5.3. <i>Yeo's approach</i>	27
4.6. SUMMARY	27
4.6.1 <i>Carey's approach</i>	27
4.6.2 <i>Hyperion approach</i>	28
4.6.3. <i>Yeo's approach</i>	28
5. RESULTS	29
5.1. EXTENDED METHODOLOGY	29



5.1.1. <i>Input</i>	29
5.1.2. <i>The Matrix</i>	29
5.1.3. <i>Output</i>	31
5.2. RESEARCH QUESTIONS	34
5.2.1. <i>Hofstede's cultural dimensions influence on global software development</i>	34
5.2.2. <i>Is a deeper localization needed?</i>	36
6. DISCUSSION	39
6.1 THE CHOICE OF WATERFALL.....	40
6.2 WEB APPLICATIONS AND SOFTWARE APPLICATIONS	41
6.3 FINAL THOUGHTS ABOUT THE EXTENDED METHODOLOGY	41
6.4. FUTURE RESEARCH	43
7. CONCLUSIONS.....	44
8. LITERATURE.....	45
9. DEFINITIONS OF TERMS	47
10. APPENDIX	49



1. INTRODUCTION

Every day you hear it on the news, you read it in the papers, you overhear people talking about it and in every single instance the word globalization seems to have a different meaning. So, what is globalization?

At a top political and economic level, globalization is the process of denationalization of markets, politics and legal systems, i.e., the rise of the so-called global economy. The consequences of this political and economic restructuring on local economies, human welfare and environment are the subject of an open debate among international organizations, governmental institutions and the academic world.

At a business level, we talk of globalization when companies decide to take part in the emerging global economy and establish themselves in foreign markets. First they will adapt their products or services to the final user's linguistic and cultural requirements. Then, they might take advantage of the Internet revolution and establish a virtual presence on the international marketplace with a multilingual website or even as an e-business.

In the context of this thesis, as described by the Localization Industry Standards Association (LISA) the term globalization addresses the business issues associated with taking a product global. In the globalization of high-tech products this involves integrating localization throughout a company, after proper internationalization and product design, as well as marketing, sales, and support in the world market.

1.1. Purpose

The purpose of the thesis is to present possible effect of cultural impact on software development.

1.2. Goal

Our main goals are to:

- Build up a extended methodology for global software development

1.3. Problem

The impact of culture on development processes and software usability is often underestimated. As we refer to in the *Background chapter* and *Established approaches* there is a need for creating new ways of developing global software

applications. The traditional models do not consider global software development and cultural aspects.

1.4. Research questions

Our investigation of the two questions is the main point of this bachelor thesis.

How Hofstede's cultural dimensions might affect the stages of global software development?

Which layers of global software application should be localized?

All software could be divided into three main categories: system software (operative systems etc), utilities and software applications. One of the important characteristics which distinguish software applications is that they are designed "...around user interaction - this interaction has a surface component known as the user interface, and the deeper component, that represents the application's logic and its core functionality." (Kersten, Kersten, Rakowski, 2001, p. 2). The fact of involvement of the user interaction in the design of the software applications makes them culturally dependent and thus interesting for our study. Examples of such products could be different types of groupware, DSS (Decision Support Systems), accounting programs. By the word global we mean that different localized versions of the same core product are sold in different countries. Compare with *Globalization (10. Definition of terms)*.

1.5. Input

Hofstede's cultural dimensions

Geert Hofstede is a Dutch anthropologist who made a large survey with 116 000 employees at IBM around the world. The results greatly influenced his model of cultural dimensions:

- Power-distance
- Collectivism vs. individualism
- Femininity vs. masculinity
- Uncertainty avoidance
- Long- vs. short-term orientation

Stages of global software development

- Analysis
- Design
- Implementation

Layers of software applications (three-tier structure)

- GUI (Graphic User Interface) layer
- Business logic
- Data storage

Global Software Development Methodologies

Creating Global Software by Carey

Global Software Development Lifecycle (GSDL) by Yeo

Hyperion solution by Livermore, Coronado

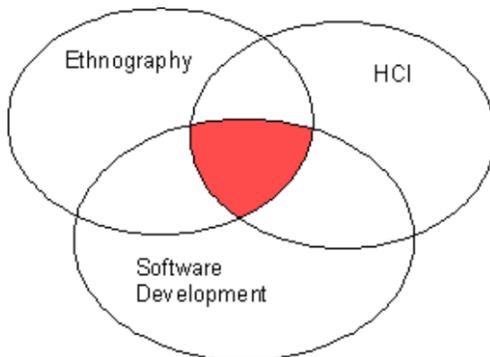
1.6. Output

An extended global software development methodology.

1.7. Target audience

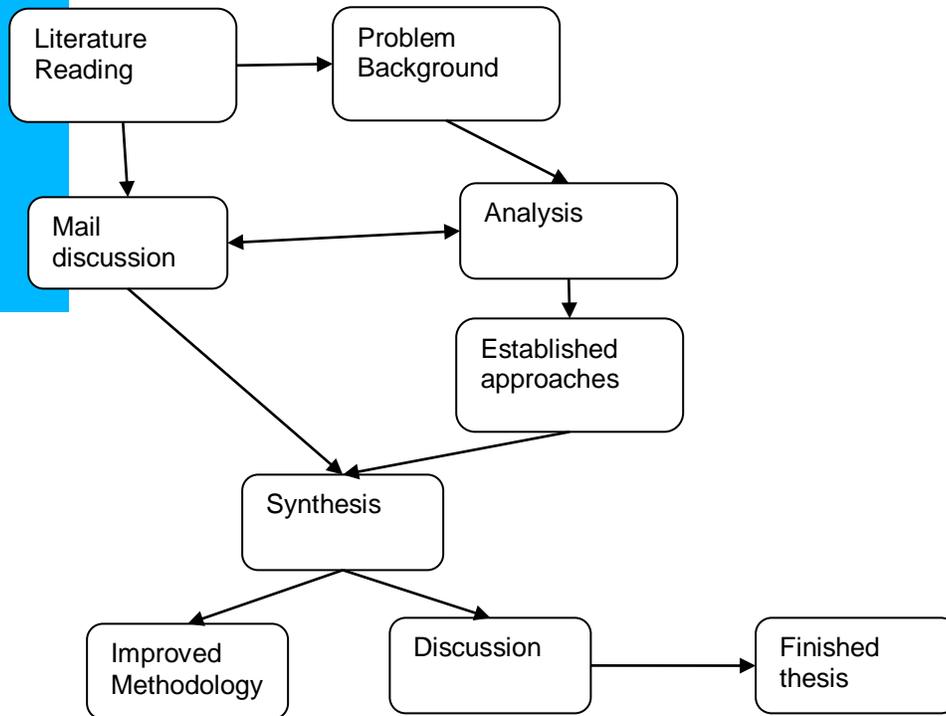
The target audience is programmers, system developers, managers, scientists, and students in the field of computers science. Furthermore, our result and conclusion should be of benefit to anyone who is interested in this area.

1.8. Delimitation



We should point out that we are not going to be able to verify our methodology practically at any development project. Under the circumstances it is not feasible. Our intention is not to produce a methodology from scratch instead we are going to extend an existing software development model.

1.9. Method



Our research was primarily based on literature reading, which included both books and articles found at ACM database. The articles turned to be an invaluable resource as they were very fresh and provided links to the authors, who were almost always willing to answer one's questions. Along with our literature study we had discussions by mail. That included both researchers and real-world developers (at Hyperion). The articles also provided us with an idea of using a cultural theory by G. Hofstede as input. By analyzing the collected information about globalization and the research domain (development of global software) we succeeded in building a ground for producing an extended methodology for global software development. Applying synthesis we proceeded and made an extended methodology (output) which was primarily based on G. Hofstede's theory, the three established approaches for development of global software, and the waterfall model (input).



2. BACKGROUND

2.1. *The importance of globalization*

For just about ten years ago the localization industry as we know it was finding a niche for itself, straddled between the software industry and translation world. Many companies have emerged as so-called specialists in managing, translating, and engineering complicated software products. (Esselink, p. 477)

When American firms began to sell software overseas, it was generally the same software that they were selling in the United States. (Esselink, p. 477). Globalization (in terms of internationalization and localization) was not considered part of software development. (Livermore, Coranado, p. 22). No modifications were made to adapt it to the language or culture of these new users, who were expected to know English and work around any other difficulties that arose. Customers in non-English speaking countries often had to wait for several months before they could purchase the product in their native languages. (Esselink, p. 477)

In today's increasingly global market, the perspective has changed. Companies cannot sell English products throughout the non-English-speaking world. And customers that before were willing to work with software localized for Americans (because it was the only available software) increasingly demanded that software would be developed with their needs in mind. (Khaslavsky, p. 456). American firms did not begin seeing the need for software localization until about 15 years ago (Esselink, 2000, p. 6). At least they started to localize software to the point that international consumers would buy it (by translating it). "When the software and localization industry matured and globalized, product lifecycles, time-to-market, and the simultaneous shipment ("simship") of different language versions became important issues for software developers." (Esselink, p. 477)

After all it makes good business sense. For many U.S. companies, the main avenue for growth is their overseas sales. "US are the largest software exporter in the world - earn more than half of their revenue from outside the US." (Yeo, 2001, p. 104). Many software designers are now faced with the task of insuring that their products are equally as usable in foreign countries as in the United States. Software companies have also realized that it is not enough to just translate their products. Companies without a global design and development process have spent hundreds of thousands of dollars re-engineering their products to meet the needs of users around the globe. "It is important, therefore, to make sure that products will work equally well in Bonn, Boston, Bogota, Beirut, Beijing, and Brisbane, which calls for creating ways to gather performance and preference data quickly, effectively, and accurately." (Dray, 1996, p. 15). "Increasing technological sophistication in many countries of the world and the resulting larger world trade implies greater need to pay attention to international aspects of user interfaces." (Nielsen, 1990, p. 291) Often this means that

companies need to learn new ways to design, translate, localize, and globalize their products. "Organizations that develop global products need to have design and development processes that consider these key requirements and differences." (Livermore, Coranado, 2001, p. 21).

2.2. Aspects of globalization

Before this discussion on globalization and the various methods used to achieve it can be continued, it is important to understand some of the areas of a software product that are affected.

2.2.1. Text representation

One matter faced during localization includes the fact that most translations from English result in longer words and phrases in the target language. "For example, when you translate a statement from English into another language, word length can sometimes increase 50 percent" (Nakakoji, p. 42). There are fewer characters in the American-English language than in the European languages with for instance the accents in the French language.

The Unicode standard has been created to address many of these character issues. Unicode assigns a 16-bit code to over 30,000 letters, symbols, and ideographs, creating a single code that is much more extensive than the 255-character ASCII standard. While the binary rendering of the capital letter "A" in ASCII would read 0100 0001, in Unicode it reads 0000 0000 0100 0001. Unicode allows for the display of multiple character sets using a single coding standard." (Unicode, Inc.)

When you translate you have to consider that extra space should be added to control buttons, menu trees and dialog boxes to account for language-specific text expansion. With web applications graphics incorporating text must be resized, as must the page area itself. "Web site designers should minimize the amount of text included in graphics and create flexible-width sites to accommodate the fluctuations in word and sentence size created by translation." (Chen, 1999, p.11-12). If the size of an interface component cannot be altered for different locales, the translator may need to alter the text, running the risk that the text message will become erroneous.

Complete translation of a natural language into another language is tricky and nearly impossible. For example, Italian has more than 500 words for pasta, Bedouin has more than 160 words for camel, and a single New Guinea dialect has 50 names for banana. We can ask ourselves how we can translate 50 names for banana into English without losing the meaning embedded in the particular representation. (Nakakoji, 1996, p. 46)

2.2.2. Data formatting

Regional Standards, which should be supported by the software include:

- Measurement formats (e.g., centimeters vs. inches)
- Number formats (e.g., different thousand separators)
- Time formats (use of the 24-hour or 12-hour clock)
- Date/Calendar formats (e.g., the mm/dd/yy format vs. the dd/mm/yy; the Chinese calendar) (ref. ISO 8601)
- Currency formats (national currencies, and related issue)
- Phone Number formats (e.g., number separators)
- Sorting rules (different alphabetical orders)

A problem that can arise with data formatting is for example the zip code that is a uniquely American concept. Other countries may or may not have postal codes and if they do, the codes may not be in the same format. Other pieces of data that are often collected in the United States, but may not be applicable in other countries, include driver's license numbers, social security numbers and mother's maiden name. A piece of software that required users in other countries to enter this information may be confusing or unusable.

Beside the traditional format and standard problems between countries there are other matters. For instance, the adoption of the common Euro currency in 2002 by members of the European Union had involved major changes in business practices and financial systems, and consequently on related software and Web applications.

2.2.3. Cultural aspects

As pointed out before is not only the language translation an important part of the localization process. The cultural aspects of globalization and in particular the user interface deserve attention as well. "It is not enough to translate the software from one language to another. It is equally important that the icons, symbols, and other cultural conventions feel comfortable and recognizable to the user." (Carey, 1998, p. 456) What you can say is that knowing the language is not enough, you must understand the culture too.

Culture can be defined differently. When we use the term "culture" ordinary we often think on the meaning "highbrow culture" for example literature, movies, music and art. The definition we use is broader and includes the beliefs, value system, norms, mores, myths, and structured elements of a given organization, tribe or society.

In explaining culture Geert Hofstede (uses the analogy of computers and programming when he says that: "Culture is always a collective phenomenon, because it is at least partly shared with people who live or lived within the same

social environment where it was learned. It is the collective programming of the mind that distinguishes the members of one group or category of people from another." (Hofstede 1997, p. 5). What Hofstede intend so say with "programming of the mind" is that every human bring with emotions, values and a pattern of behavior. Such things that are inherited and learned mostly from our childhood but all together with our culture make us to the person where we stand today. Of course this does not mean that humans are programmed like computers!

Some aspects of culture can be readily seen on the surface and are pretty obvious, like currency and time (see *Data formatting*), clothes, food and architecture. However, if you look beneath the surface you can see some other aspects (e.g. "deep culture").

They could include:

- Perception of what is right and wrong
- Business etiquette and protocol
- Political tradition and symbolism
- Family and social interaction
- Role of religion
- Literacy and learning style

(Flanagan &, p. 75, 1999)

The innermost dimensions of culture can be unspoken rules and are outside of conscious awareness which makes it all a little bit harder to identify and study.

What we can learn from this for instance is that cultural-dependent or ambiguous symbols: as a rule, body parts (e.g., hand gestures), religious symbols, graphics with more than one meaning and cultural- specific symbol (e.g., the stop sign) should be avoided, as they may lead to misunderstandings.

"Different cultures have different psychological associations for color, for example. Red, a color indicating danger or warning in many cultures, signifies celebration in China." (Nakakoji, 1996, p. 43) As described color, flow, shape, etc. are important factors to consider.

Adjust and enable the appearance and orientation to account for national or cultural differences. For example, using a post letterbox as an icon for e-mail may require different images for different countries. For example, these images from a prototype for Sabre's Planet Sabre, one of the world's largest extranets, whose UI+IV was designed by the authors' firm, shows variations of mailbox icons to account for national differences (see Figure 1)



Figure 1: Planet Sabre prototype international mailbox icons

- As a universal sign set reference, consider using as basic icon/symbol references the signs, or signs derived from them, that constitute the international signage set developed for international safety, mass transit, and communication.



Figure 2: International pictograms from Pierce, 1996.

- Avoid puns and local, unique, charming references that will not transfer well from culture to culture. Keep in mind that many "universal" signs are covered by international trademark and copyright use, e.g., Mickey Mouse and the Smiley smiling face. In the USA, the familiar smiling face is not a protected sign, but it is in other countries.
- Be aware that office equipment such as telephones, mailboxes, folders, and storage devices differ significantly from nation to nation. (Marcus, 1999)

3. HOFSTEDE'S CULTURAL DIMENSIONS

One of the most interesting studies concerning culture and management is the one by Geert Hofstede. It is based on interviews with 116 000 employees at IBM around the world. The results of the survey, covering employees in 72 national subsidiaries, 38 occupations, and 20 languages, and at two points in time: around 1968 and around 1972, put a ground for his model of culture, consisting of five dimensions. In the context of this thesis a cultural theory is a necessary input. According to the definition of global software (*10. Definition of Terms*) human-machine interaction is a very important part of software development. The developer decides how that interaction is to be designed. When Hofstede speaks about intercultural communication he emphasizes three phases to be very important awareness, knowledge, and skill. Those phases are as important for developing software and designing interaction as software communicates. According to Hofstede the awareness is to be conscious or aware of the relativity of one's own mentality (or mental software as Hofstede puts it). Knowledge means that one should learn the cultures one is to deal with. The skill is built on the awareness, knowledge plus exercise or praxis.

3.1. Hofstede's dimensions

According to Hofstede, cultural differences can be explained by a five-dimension system.

- Power distance
- Collectivism vs. individualism
- Uncertainty avoidance
- Femininity vs. masculinity
- Long- vs. short-term orientation

3.1.1. Power distance

According to Hofstede, the power distance (PD) is basically how much the less powerful members of organizations and institutions in a country expect or accept that power is not divided equally. By institutions we mean such elements of a society as a family, a school, etc. The power distance describes the grade of dependency of people with a lower position on people "in charge" in a culture. In countries with low PD employees prefer bosses to consult and explain the background of the decisions and orders. People are bound to feel that there is mutual dependence between them and the executives. In countries with high power distance the employees are more dependent of their bosses and expect them to give orders.

An interesting observation is that differences in PD values in different country were almost due to differences in the values of the middle-class part of the participants. People with just some vocational training or uneducated people had almost the same average attitude to PD- high PD. They preferred authoritarian leadership.

According to Marcus and Gould (Marcus, Gould, 2000) PD may influence the following aspects of the design of a web UI:

- Highly/less highly structured access to information. Refers to high/low PD.
- Strong/weak focus on expertise, certificates, official logos and stamps.
- Prominence given to leaders vs. employees/citizens.

In the examples presented one can see two different web pages of two universities in two countries, Russia and Sweden. Sweden is ranked 47/48 of 53 countries when considered PD. Russia has not been ranked by Hofstede but we felt it was appropriate to use Russian pages for comparison. Russia certainly is a country with very high respect for power distance (slavery was abolished as late as 1861!). The Stalin's brutal rule is not known for encouraging of discussing of leaders orders. The soviet leaders after Stalin are not either famous for letting people to make their own decisions.

When looking at the Russian web pages it is very easy to recognize the focus on official logos - the Russian state symbol. The menu is structured in a way that the "students" link is almost the last one you see (second from the bottom). On the contrary the Swedish page put pictures of students on the front page (not the authorities). There is no doubt that the Swedish page gives a feeling of being more informal and students-oriented (prominence is given to students). The Russian page is more rigor, prominence is given to the authority (e.g. teachers, rector).



Figure 1. St. Petersburg University's main page. <http://www.spbu.ru>



Figure 2. Karlstads University's main page. <http://www.kau.se>

3.1.2. Individualism or collectivism

These dimensions describe how much the individuals deploy. In individualistic societies the relations between individuals are not very strong. Everyone takes care of themselves and their next of kin. Collectivistic cultures are characterized by strong family relations, strong group relations and networks. Members of an individualistic society focus on self-respect whereas members in a society with collectivism give priority to fit into the group in a harmonic way. It is important to emphasize that the word collectivism in the context does not mean the state's power and influence on the individual but the group's.

Marcus and Gould believe that collectivism and individualism may too affect some aspects of the web design in the following ways:

1. How images of success are presented- through materialism and consumerism vs. achievements of social-political goals
2. Prominence given to youth and action vs. maturity, experience

The examples the authors offer are based on what is put in the center of two national parks' web sites. The first site made in US is focused on the visitor and her/his goals (an individualistic culture), while the second put the nature in the center (Costa Rica- collectivistic culture compared to US) downplaying the visitor.

We think that company slogans one puts on the net could be influenced by this dimension. "Think different" by Apple gives an impression of being a very individualistic slogan which emphasizes the importance and value of individual contribution and integrity. One feels to be encouraged to differ from others.

3.1.3. Uncertainty Avoidance

Hofstede defines this dimension as how much (in what grade) members of a particular society feel uncomfortable with or threatened by uncertain, unfamiliar situations. This feeling expresses itself in stress, a need for predictability, written or unwritten rules. Thus some cultures are more anxious and others. Hofstede remarks as well that people belonging to cultures with high UA (more anxious) are more expressive, they communicate with their hands, it is accepted that one raises one's voice, etc.

Marcus offers the following ways of application of this dimension on web design for high UA cultures.

- Simplicity with limited amount of choices
- Navigation schemes intended to prevent users from becoming lost

On the contrary when developing for low UA cultures one may use

- Complexity with maximal amount of choices
- Less control of navigation
- Help system might focus on understanding underlying concepts rather than narrow tasks.

Furthermore we would like to add that as Hofstede notices cultures with different value of UA differ in their attitude to the innovative and the different. That means that one has different grade of freedom designing user interface for different countries.

3.1.4. Masculinity-Femininity

According to Hofstede there is a tradition of women performing mostly at home and men performing outside household. Hofstede suggests that it could be explained by the order which is thousands of years old. The order basically was due to the fact that children needed their mothers to be close to feed them and to take care of them. At the same time men didn't need to always be close to succeeding in supporting the family. Hence the tradition of men expected to be self-assertive, tough, and competing and women expected to be engaged with household, children, and relations in general. Thus, masculine societies are characterized by sharp gender roles: men are self-assertive, tough, and seek for tangible success. Women are supposed to be unpretentious, modest, and engaged with life quality. In feminine



societies on the contrary the gender roles overlap, both men and women are expected to be modest, tender and aim at life quality. For the Swedish readers it may be interesting to know that Sweden is the most feminine of the 53 countries ranked by Hofstede.

According to Marcus and Gould the cultures with high MAC value (Japan, Austria, Italy, and Mexico) appreciate following aspects which may influence UI design:

1. Traditional gender/family distinctions
2. Work tasks and roles
3. Attention gained through games and competitions
4. Graphics, sound and animation used for utilitarian purpose

On the contrary the feminine cultures would emphasize:

- Blurring of gender roles
- Mutual cooperation and support rather than winning
- Attention gained through poetry, visual aesthetics

3.1.5. The long or short term orientation (LT)

This particular dimension is a very interesting and describes one of the fundamental differences between "west" and "east". Hofstede contends that while western people seek truth, eastern people seek virtue. When western people talk about religion they are interested in which church one belongs to, what one believes in (faith). In east people are interested in what rituals one carries out, how one lives one's life.

According to Hofstede the West believes that there is one ultimate truth and one's faith matters. The East believes that there are a lot of truths and that one's lifestyle matters, not one's faith.

Among the aspects of lifestyle that Hofstede highlights are thrift, perseverance, and respect for social obligations. People belonging to cultures with (relatively) higher value of LT (China, Hong Kong, Taiwan, Japan) tend to respect social obligations within reasonable bounds, are thrifty or economical with resources, and persevering working for results in same distant future.

People belonging to cultures with (relatively) low LT respect social obligations at any price, tend to look as wealthy as neighbour even if it leads to one spending more than one can afford, and finally want fast result of their work.

Marcus suggests that that when one develops for countries with high LT one should emphasise the following:

- Content focused on practice and practical value
- Relationships as a source of information and credibility

When a country with low LT is the target:

- Content focused on certainty and beliefs
- Rules as a source of information and credibility
- Desire for immediate results and achievements of goals

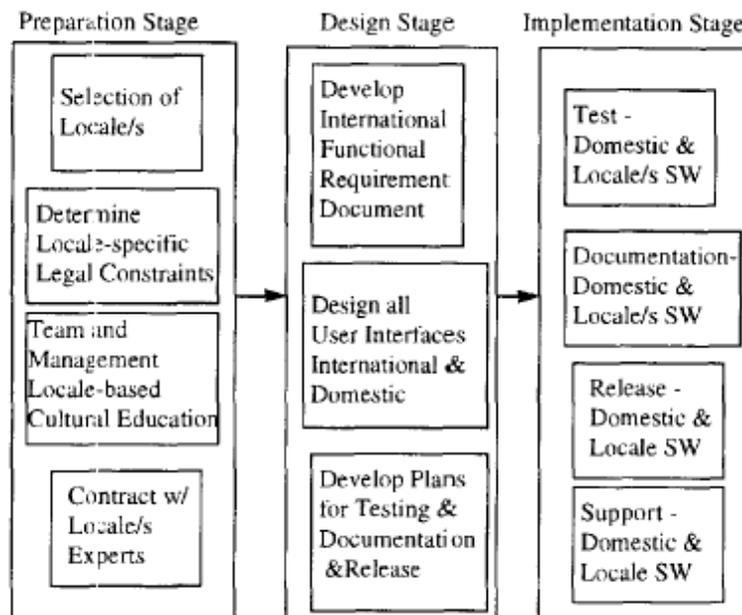
This particular dimension is the most obscure when it comes to identifying it in some web sites' design. The examples provided by Marcus we find to be not very persuasive. Nevertheless we have some suggestions about how customers/users culture could affect the process of development global software. We present those suggestions in chapter 5, "Extended Methodology".

4. Established approaches

During the work we came across quite a few sets of guidelines for global software development. Some of them were more complete than others. In the end, we chose three of them, which appeared to be comprehensive enough for our purpose.

Carey's approach

Carey suggests a structure which follows the traditional life cycle model and includes the three stages: preparation (analysis), design, and implementation. If one examines the picture of the model then an interesting question arises. There, namely is not any core functionality design included in the model. The design stage consists of development of the functional requirement document, design of GUI, and plans for testing and documentation.



Adapted from Carey (Carey, 1998, p. 456).

The Hyperion Solution

Jose Coronado and Carrie Livermore are managers of the Localization Team at Hyperion (www.hyperion.com). Hyperion is a US-based company which produces software for financial analyze, performance management, business modeling. Most of the customers are big corporations such as Coca Cola, CityBank, and Siemens. Hyperion has offices in 20 countries. Coronado and Livermore define some "basic" internationalization problems, these are: product internationalization and localization not considered as part of the development, user interface engineering non-existing as a career path. (Coronado, Livermore, 200, p.22). Probably the authors should have emphasized that these are basic internationalization problems for companies of the size of Hyperion (yearly revenue more than half a billion dollar).

Yeo's approach

Yet another approach is Global-Software Development Lifecycle (global-SDLC). Alvin W. Yeo, researcher at University of Waikato (New Zealand) has conducted a localization project and described his experiences and the methodology (Global-Software Development Lifecycle) in an article ("Global-Software Development Lifecycle: An Exploratory Study"). The author writes about global-SDLC as if it were an established methodology of developing global software. We asked the author how popular the methodology was. According to his answer (Yeo_020404) the methodology was used by Microsoft. Yeo contents that the cycle consists of three phases and those are design, implementation and evaluation.

4.1. Organizational issues

4.1.1 Carey's approach

According to Carey (Carey, 1998, p. 456) there are following options for the organization structure:

1. one team working with both US and internationalized versions and another team responsible for localization
2. one team working with both US and internationalized versions then each local office in different part of the world is responsible for localization for their region
3. one team responsible for the US version and another team working with internationalization and local releases.

Of course we believe that this strict division is not always applicable. Especially if the company in question is not so big then this approach may not be feasible. One of us experienced worked with localization in a firm with 5 programmers, there everyone did a little of everything.

4.1.2. Hyperion solution

The authors of the article focused a great deal on organizational issues. They emphasize the fact that Hyperion had to build new teams to handle globalization of



its products. They emphasize the importance of the executive support. Globalization even became one of the other aspects which affect bonus compensation structure in the company.

Hyperion created new (for the company) careers: user interface engineers (responsible for the interaction design, task flow, screen layout), visual designers (responsible for the visual elements, like icons, splash screens, product banners) and usability engineers (responsible for collecting user-centered data, like task analysis, usability testing).

Usability engineers issue interested us so we asked a question about it to Hyperion. We wondered in particular what methods they used for usability testing. As an answer we received a high level inventory of usability engineering methods used at Hyperion. Those methods are:

- Usability Evaluation Sessions
- Remote Usability Evaluation
- Cognitive Walk troughs
- UI Internationalization Review
- User Interface Design
- User Interviews
- Task Analysis
- Heuristic Evaluation (Usability expert review)
- UI Consistency Reviews
- UI Standards
- WEB UI and GUI Design Training

(Adapted from Coronado_020328)

Using Carrie Livermore and Jose Coronado's kind assistance we, among other things, tried to check for some issues Mr. Yeo (Yeo, 2001) considered to be important. To name a few of them: usability evaluation using participants from the target culture, to use UATs which are appropriate for the target culture (read different UATs). According to the article User Centered Design (UCD) engineers developed design guidelines for the web and GUI.

According to Coronado and Livermore one of the most important findings Hyperion did was that one has to have a project manager in charge of the localization project. Both the team working on localization and external vendors (translators) need a point person to interact with. The company among other things also came to the conclusion that employees with language knowledge are better used as reviewers and testers of the product, after it has been translated.

Another important issue is the one of understanding of the importance of the internationalization by the management. Livermore and Coronado stress the fact in the article. We wondered a little what kind of management exactly they meant and how that understanding could be achieved. Carrie's answer was that they meant

"...executive managers, the people with the real power. They came to this understanding not only by our educating them, but also by seeing the revenue numbers that were gained by overseas sales."(Livermore_020416).

Jose added as well that even "...the mid-level managers that are closer to the product are also an important part of the support and success of a usability oriented activity. If middle level managers support and allocate development time to address issues and findings, then you are effective. Otherwise, you put the "check mark" in the activity but you do not produce a measurable result."(Coronado_020416)

4.1.3 Yeo's approach

No suggestions.

4.2. Cultural issues

4.2.1 Carey's approach

Carey points out the importance of understanding the target culture. She outlines different strategies for doing this. Carrie suggests following:

- hire an outside consultant who already understands the target culture. The consultant should preferably speak both the target country language and the language of the country where development is being done.
- make a cultural study by sending human factors experts to the country
- hire members of the target country

As we described in the "Background" the term culture is rather complex and asks for definition of what one means with "understanding a culture". Our discussion with managers of the Localization Team at Hyperion gives us an interesting idea of that "cultural consultant" may be a person at marketing department. Such person would have users' feedback and naturally could consider which cultural improvements different products need. Of course such a person should speak both the target country and the development country languages.

Similar thought are found as well in one of Susan Dray's articles (Siegel, Dray, 2001, p.19, 20). She found out that Marketing department of the organization she was involved in a project with was very much interested and engages in issues like interaction design. As interaction design is extremely important when considering cultural aspects impact on global product usability, we think Dray's observations are very useful.

4.2.2 Hyperion solution

The cultural issues are a very important part of the thesis so we decide to put some

questions about them to Carrie and Jose. We were especially interested if methods for usability evaluation differed for different countries and if tests were carried out in target countries. As Jose answered us the test methods varied a little between countries but not significantly. Hyperion does test that products are compatible with different types of hardware: "We make sure that we use hardware (i.e. UK English keyboard vs. US English keyboard) from the location we are at." (Coronado_020328). That is an important issue highlighted by Esselink (2000) as well. Interesting news was that Hyperion conducts usability testing only for its English speaking prototypes: "Unfortunately, we do not do usability testing in our localized versions of the software. We only use English version prototypes or products for these sessions. The requirements for different languages would demand from us to use native speakers as facilitators as well as hardware with more specific requirements for the different locales." It sounds like it is too much trouble with usability testing for localized versions. The tests are though conducted in the target countries and even: "More than target countries, they are conducted with target users, target sector companies and target regions. Target users: Application Administration vs. End User Target Sector. Companies: Banking or Financial, Government or Public Sector, Big 5 Consulting Firm, Insurance, etc. Target Regions: North America (US only or US and Canada), Europe, Asia Pacific and Latin America. These last two regions traditionally are not targeted as heavily as North America and Europe." (Coronado_020328)

As one can see Hyperion addresses the language related issues mostly, while other cultural aspects may be left without any special concern. We thought that it might be justified by the general profile of the users of Hyperion's products, but as Carrie told us Hyperion didn't do any general user profile identifying.

4.2.3. Yeo's approach

Coding of the software is recommended to be carried out with some guidelines in mind. One should namely use the big organizations (like Apple, HP, Microsoft) recommendations. Which particular company's recommendations are relevant in a particular case depends, of course, on the target platform for development. The author complains again about companies taking care about mostly one aspect of cultures, namely "Language and its language-associated issues, such as character sets, sorting, character display, and the data display formats (date, time, currency and address formats)." (Yeo, 2001, s.105). Yeo claims that the rest (issues like color associations, forms, icons and metaphors) is neglected. In this particular context it seems weird that the researcher himself didn't try much to address these cultural issues when conducting internationalization project of his own.

4.3. International interface

4.3.1 Carey's approach

The author provides developers with following guidelines:

- avoid complex design,
- use dialog tab controls to limit the length of menus,
- use color carefully,
- display warnings or messages until the user dismisses them,
- draw text on bitmaps independently,
- allow for customization of keyboard accelerators and shortcut-key combinations,
- use on-screen keyboards, and
- use standard windows controls and mechanisms.

These guidelines are very useful and some more could be found in a book by Esselink (Esselink, 2000).

4.3.2 Hyperion solution

In the area of International Interface there is very much done at the Hyperion. The company has developed a set of standards and guidelines for GUI development. According to Jose "Standards are perceived as an imposition into the development team. They tend (according to some developers) to stifle and limit their creativity. The enforcer of the standards may be perceived as the "GUI Police."". Standardization has its cost and as we can see some developers could be frustrated over their creativity being controlled by the regulations. The guidelines are more like suggestions not like impositions.

4.3.3. Yeo's approach

The design phase includes internationalization and localization. The author refers to Carey's conspectus for guidelines for design phase (Carey, 1998). Further Yeo suggests a design evaluation to be done, he does not although describe which of the design artifacts should be evaluated and how. That is especially strange, because the author had carried out a globalization project himself, when producing his paper. To clear up this issue the author was contacted and asked which artifacts of the design were evaluated in his project. Yeo answered that mostly the evaluation was conducted to ensure that language related issues were covered. The fact that only language specific part of the design is evaluated seems to be awkward. We consider the cultural issues (discussed in the background) to be as important as language.

4.4. Documentation

4.4.1. Carey's approach

Carey emphasizes the importance of the consistency of the used terms. The proposed steps are:

1. planning
2. glossary development
3. writing
4. creating the artwork
5. translation
6. quality control
7. production of final copy

The translation of the documentation is a process more difficult than it might seem. Hyperion (the company named it previously) actually outsource this particular process, very much due to the importance of consistency of the used term. Hyperion emphasizes that consistency of terminology is very valuable even at a higher level than the one company's products.

4.4.2. Hyperion solution

The article mentions that the staff had some training at how to address some common problems writing documentation for global markets. As consistency of the terminology is a very important issue Hyperion actually has built a database of terms and as Carrie describes it "Believe it or not, it was a problem to get each team to agree on certain terms!" (Livermore_020423).

4.4.3 Yeo's approach

No suggestions.

4.5. Testing, QA

4.5.1 Carey's approach

"The process of quality assurance takes time and money, but it is mandatory and can easily make or break the product" (Carey, 1998, s. 461). As the citing says it is very important to plan and to do QA testing and as this process may come to involve test participants from different cultures it is extremely important to assure that the techniques used are appropriate. Later we are going to present some examples of using inappropriate (due to the participants' culture) techniques.

Carey offers a plan for quality assurance and emphasizes that QA is "one of the most important aspects of localization" (J. M. Carey, 1998, s. 460).

An international quality plan according to Carey:

- overview of product (functions, target audience, level of international enabling),
- overview of QA test plan,
- purpose and scope of testing,
- exclusions,

- description of testing approach,
- responsibility for enabling QA (core vs. international engineers),
- information on external testing (beta test sites, possible bug hunts to classify severity and recourse)
- resources required (people, hardware, software, OS),
- high-level view,
- general schedule, and
- quality expectations.

The phase "bug hunt" is a very interesting one. According to Carey's description the activity is seeking for bugs carried out by developers who are not part of the team, which produced the software. It is interesting that Carey even names how much they get paid for every bug found (5\$-10\$).

4.5.2. Hyperion solution

Quality assurance is one of the key issues if one would like to keep one's customers. Hyperion sets QA high at a priority list. The company has experiences that say-different cultures have different acceptance/ tolerance of defects in product. As Carrie (a manager of Localization Team at Hyperion) says "Quality to the Japanese people is perfection. There must be few or no mistakes in the product, especially when it comes to translation quality. You only get one chance in Japan to show your product. If it has many defects, or the translation quality is not good, they won't look at your product again for many years, if ever. The first impression lasts a long time. I think Americans are much more accepting of defects. It is known that software is not perfect and we accept product with a few bugs." (Livermore_020507)

4.5.3. Yeo's approach

Here a very interesting question that is brought up for discussion; how the usability evaluation techniques (developed in West) are suitable for other cultures (like for example Asia). The Usability Assessment Techniques (UATs) tested by the author were: think aloud (TA), System Usability Scale (SUS) and interview. While the TA session the participants express their feeling while completing some assigned tasks. SUS is a kind of a questionnaire; the participants fill in scores for some system qualities. The System Usability Scale is usually used after the respondent has used the system, but before any discussion takes place. Respondents should be asked to give their immediate, spontaneous response to each item of the questionnaire, rather than thinking about items for a while. (Brooke, 1986).

4.6. Summary

4.6.1 Carey's approach

Carey's article covers many of the most important issues of global software



development. The author offers detailed plans for documentation and QA phases. Carey's suggestions about cultural education of the staff involved in development of global software products seem to be supported by all the other sources we used. We like the way that the article is structured and we actually tried to apply a similar structuring to our description of another approach, that by Hyperion. The next approach is by a Malaysian researcher who actually refers to Carey when it comes to a description of the design phase.

4.6.2 Hyperion approach

The Hyperion approach is a modern and approved way of producing global software. The article and the discussions (carried out via mail) provided us with a really rich input for our extended methodology. The information that we acquired from Hyperion is particularly valuable due to the fact that it comes from a company which implements and elaborates methods and strategies for making global software for a wide variety of countries, such as Japan, China, US, France, and Russia.

The Hyperion approach and Carey's approach have similar attitude to many important issues. Hyperion gathers International Requirements, emphasizes the importance of testing products with real end-users, and has a special database of terms to ensure that the terminology used is consistent. Some other phases the Hyperion solution suggests to be treated differently. For example the task of translation is outsourced according to the Hyperion solution.

4.6.3. Yeo's approach

After completing the usability evaluation the author checked if all the three testing approaches' results were consistent. He found that participants who expressed frustration and some negative feelings during TA (more spontaneous activity compared to filling in System Usability Scale) gave the software high score in System Usability Scale and interview. This finding implies that the usability techniques which worked fine in West may not perform as good in the Asian culture (Malaysia). To explain a little what was wrong we could say that in the best case all the three approaches' results should be very consistent. If they are not, then- by some reasons- the participants changed their mind about the same issue/question during the session.

An interesting finding was that the responses of the participants, who were either familiar with the author or experienced in computer use, were more consistent. Yeo provides an explanation for that which he bases on Hofstede's theory of dimensions of the cultures. Malaysia is ranked 36 of 53 countries when considered collectivism vs. individualism by Hofstede (Hofstede, 1994). That means that Malaysia is a rather collectivistic culture, compared to US (ranked 1 of 53- very individualistic culture). The author's comment is as follows: "This inconsistency may be attributed to the fact that the participants did not want to comment negatively about the spreadsheet (the software that author produced during his project) 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." (Yeo, 2001, s.105).

Finally we would like to summarize the most important issues highlighted by the paper by Yeo. A particular culture's value or rank in Hofstede's classification may affect the possibility to apply particular UATs (Usability Assessment Techniques). One is recommended to carry out a pilot study to see if the UATs one intends to use work in the target culture, if the culture is not known previously.

5. Results

5.1. Extended Methodology

5.1.1. Input

As an input or base for our extended methodology we use the three methodologies described in chapter 2, Hofstede's cultural theory, and the Waterfall model.

5.1.2. The Matrix

In the following matrix we try to present the three established approaches and an extended methodology of our own. The extended methodology shall cover all the important aspects of global software development that the established approaches fail to take care of.

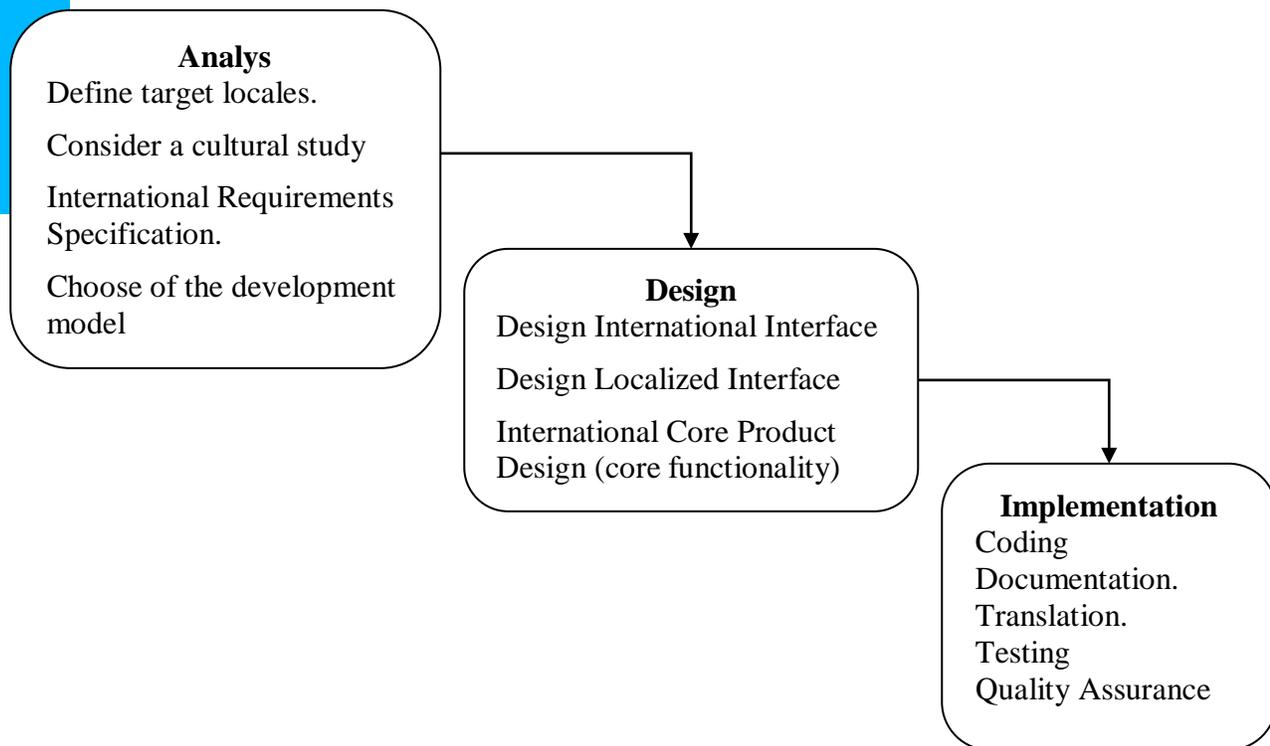
Phases	Carey	GSDL	Hyperion	Extended Methodology
Define Target Locales	Yes	Yes	Yes. Localization staff gets training; Carrie regularly attends conferences on localization.	Yes, consider if a cultural study is feasible and meaningful.
Requirements	International Functional Requirements Document/Cultural Education of the staff	No suggestions	International Requirements	Gather International Requirements
International Interface Design	Design all the Interfaces: domestic and international	No suggestions	International Interface, not much consideration about cultural impact. There is a set of guidelines.	Take Hofstede's cultural dimensions in consideration. Involve Marketing people.
International Core Design	No suggestions about core functionality design	No suggestions	Differences in business processes considered to be very important to support.	Depending on the type of software. Hofstede's dimensions may be useful in modeling interaction between the user and the software
Coding	Is not affected	Is not affected	Is not affected	Is not affected
Documentation	An extensive set of guidelines for writing of documentation	No suggestions	Consistency of terminology, a special database of terms is build.	Carey's set of guidelines.
Translation	Do not outsource	No suggestions	Outsource	Outsource
International Usability Evaluation	Suggestions about a plan for International QA and testing	Chose UATs carefully. Make a pilot study to see if the UATs will work? Localized	US product/ end users. Techniques do not differ much.	Evaluation methods may need to differ for different countries



<i>Phases</i>	<i>Carey</i>	<i>GSDL</i>	<i>Hyperion</i>	<i>Extended Methodology</i>
		product/ end users		

5.1.3. Output

We follow the traditional three steps approach in presenting the extended methodology; the steps are Analysis, Design, and Implementation.



Analysis

- Define target locales.
 - Output: you should know which the target locales are.
- Consider a cultural study.
 - Output: Cultural reference/model should be completed and decision whether a cultural study is feasible should be made.
- International Requirements Specification.
 - Collect requirements, study them and see if they are not too different for different customers (countries) to proceed and design one core product.
 - Output: The specification is approved by the customer (the customer may be the Marketing Department representative).
- Choose of the development model most suited for the project.

Based on project objectives choose a model most appropriate for the project at hand. The customer's culture may affect your choice. For example as

oriental cultures are long-term time oriented your Chinese customer may be less inclined to demand for early prototype (provided he/she knows exactly what is to be developed).

Design

- Design International Interface.
 - Add approximately 30% extra space to each control in the dialogue boxes as text strings can be longer when they are translated so it is important to leave adequate room for expansion in.
 - Avoid culture-dependent references and cultural-specific symbols (e.g., the stop sign hand gestures, religious symbols, graphics with more than one meaning) should be avoided, as they may lead to misunderstandings.
 - Be aware that "everyday" items such as telephones, mailboxes, calendars, folders, and storage devices differ significantly from nation to nation.
 - Output: the core (international) product's GUI should be completed.

- Design Localized Interface.
 - Follow reference guidelines for good colour usage and alter color schemes if necessary.
 - Respect national and cultural variations in colours, where feasible, for the target markets (as they can have different meaning).
 - High PD¹ (and high UA as well) structure information more strictly when designing menus. The depth of the structure is going to be deeper demanding more steps to come to certain functions but the number of alternatives presented at the same time is going to be fewer.
 - Low MAS: consider development of "feminine" set of controls (e. g. buttons, forms etc. with non-rectangular corners and with more aesthetic appeal) by using polymorphism. That particular suggestion should probably be explained more thoroughly as the word "feminism" has many definitions. In the context of this thesis we use Hofstede's definition (blurring of the traditional gender roles) and we use some of Marcus and Gould's (2000) ideas about the implications of that particular dimension on design of web user-interface. Thus a feminine GUI is for us an interface with a strong aesthetic appeal when considered for example choice of colors, forms of the controls, choice of sounds, and such visual elements as pictures.
 - High UA: consider using "wizards" for functions that demand more than two clicks in menus.
 - High UA: be careful introducing new concepts or controls into GUI, members of the societies with high UA consider the different to be

¹ PD=Power distance Index, UA=Uncertainty Avoidance Index, MAS=Masculinity Index, see chapter 3, *Hofstede's cultural dimensions* and Appendix 1.

dangerous. On the contrary designing for country with low UA feel freer to introduce something new.

- Output: the set of localized GUIs should be completed.
- International Core Product Design (core functionality).
This stage is the most challenging and interesting. Depending on the type of application being developed an internationalization of the core of the product may be needed or may not be needed.
 - Reexamine Requirements Specification and interaction models (made during the GUIs design). Decide if any core functionality localization is needed and feasible. If so proceed to the next step, otherwise proceed to Implementation.
 - Design Localized Core parts/libs
 - Output: Core functionality design is completed. If necessary- the design is made for international core and localized of the core.

Implementation

- Coding
 - Output: the product should be developed and ready to be tested and debugged.
- Documentation.
A very important part. Here the product documentation (e. g .on-line help etc) should be created.
 - Keep sentences short - use as few clauses as possible.
 - Always speak in the active voice.
 - Try to maintain present tense all the time.
 - Avoid synonyms- agree upon the terms used throughout all the teams in the company.
 - No jargon, sayings
 - Create and include a glossary of terms which could be used by users in case of confusion.
 - Output: the documentation should be written and ready to translate.
- Translation.
 - Outsource translation.
 - Output: the documentation should be translated and ready to evaluate.
- Testing. The artifacts that are to be tested are the code (functionality), quality of translation, and the usability of the product.
 - Testing of the functionality
 - Document the bugs and fix them (go back to stage code or even Core design if necessary).



- Testing of the translation of the documentation
 - Translate random pages of the documentation back to the "default" language and compare with the original.
 - Let employees with the knowledge of the target languages to read and review the translated documentation.
 - Fix the errors.
 - Usability testing
 - Use localized products with end-users.
 - Use UATs which are known to work for the target locales.
 - Document the flaws and fix them if possible, or fix them in the next release.
- Quality Assurance, assure that the product has an expectable quality level set for the target locale.
- Different countries have different defect-tolerance. (For an example read chapter 4.3.5.)
 - What the "expectable quality level" is could be an issue for another thesis and it is kind of standard that companies form themselves. An example could be number of bugs per 1000 lines of code or per 1000000 lines of code.
- Output: a ready to ship product.

5.2. Research questions

5.2.1. Hofstede's cultural dimensions influence on global software development

Research question:

How Hofstede's cultural dimensions might affect the stages of global software development?

As described in chapter 3 Hofstede's cultural dimensions are:

- Power distance (PD)
- Collectivism vs. individualism (CI)
- Femininity vs. masculinity (MAS)
- Uncertainty avoidance (UA)
- Long- vs. short-term orientation (LT)

During our investigation we succeeded to find out how they may influence development of global software. Different dimensions actually can affect different phases of development. The stages that could be of interest in the context are:

- Analysis
 - Choice of the development model most suited for the project
- Design
 - Localized User Interface Design
 - Core Functionality Design
- Implementation
 - Usability testing

Choice of the development model

The choice may be influenced by the LT dimension. Suppose your client is from China which is a country with high LT. In that situation he/she might not demand fast, tangible results from you (a working prototype). Your Chinese customer may feel comfortable with the idea of waiting for the product for a while. In case that you are working for an American client the situation may be the opposite.

Localized User Interface Design

After separating the cultural variables in the International User Interface Design you adapt that interface to the target country by localizing it. The following dimensions are of importance here: PD, MAS, and UA.

Like presented in our extended methodology we suggest for example a more “feminine” set of controls to be used for a Low MAS culture, and structure information more strictly when designing menus when developing for a country with high PD value. By structuring menus so that for example at most 5 alternatives are visible at the same time you can reduce uncertainty for the target user (for high UA). Developing for a country with high UA consider using a “wizard” in cases of functions demanding more than two clicks to reduce uncertainty.

Core Functionality Design

The following dimensions can affect this particular stage of development: UA and PD. As an example we can mention that during one of the UCD activities Hyperion discovered that the users in the US expected a lot of flexibility being offered by some of the business processes supported by the application being tested. At the same time the German users would like to see "a more strict approach" (Coronado, Livermore, 2001, p.23). According to Hofstede's ranking of the 53 different countries there is a very big gap between the US and Germany. Germany's rank is 29, while the US's are 43, which means Germany is much more "uncertainty avoiding" than the US are. Could that explain the difference in expectations? We believe that it could and it may provide some theoretical base for Hyperion's real life findings.

Usability testing



An interesting finding we found while reading about Yeos (2000) work was that he made a study in Malaysia about usability and found out that the participants in the experiment did not want to comment negatively because they were afraid that it would cause the experimenter to lose face so they gave positive comments instead. Yeo comes up with an explanation that he bases on G.Hofstede's theory of *Collectivism vs. individualism*. According to Hofstede's scale (Appendix. 1) Malaysia is ranked 36 of 53 countries when considered collectivism vs. individualism. That means that Malaysia is a collectivistic culture and is maybe an explanation to why the usability techniques which worked fine in West may not perform as good in Asian cultures like Malaysia and others, and you should therefore consider using another technique. If we summarize up we conclude that a particular culture's value or rank in Hofstede's classification may affect the possibility to apply particular UATs (Usability Assessment Techniques).

5.2.2. Is a deeper localization needed?

Research question:

Which layers of global software application should be localized?

As people of different cultures plan and carry out their business activities in different ways- the artifacts that support those activities should be designed and build with these differences in mind. We asked Jose Coronado (Localization Team manager at Hyperion) about how culture may influence global products design at Hyperion. His answer shows that User Design Engineers at Hyperion take cultural differences into consideration while building GUIs. The Localization Team focuses primarily on "understanding the business process and how it varies from region to region, or from industry to industry. This is probably the area where the understanding of cultural differences is applied in the design of complex business analysis products."

If accounting applications are considered then it depends on the laws and practices of the target countries if there should be some differences in functionality. This type of global software applications are developed with different modules/components to account for differences of the laws of different even today. Hyperion is a good example.

Another type of applications, for example a DSS (Decision Support Systems) or a spreadsheet with a help agent embedded, is something not explored so well yet. Suppose that the agent is to teach users to master the software. How should the agent interact with them: as a guru or as a friend? In western societies a teacher is looked upon as a friend, in eastern- as a father (one of Hofstede's dimensions, Power Distance). The cultural differences may demand for different interaction styles.

Culture

Culture is a set of shared values, beliefs and behaviors. The question here is

whether there are any general cross-cultural laws of culture? According to the authors there are two different opinions on this matter.

- The first, holistic point of view tells that there is no universal culture nor universal laws. One cannot break down a culture into a set of pieces or components in order to understand it (like one can do with a machine).
- The second, reductionist approach assumes that a culture is a set of symbols which follows some general structure and can be broken down and understood in the same fashion as a language.

The first (holistic) perspective is the one we base our methodology on.

The authors offer a more sophisticated view on culture, namely that any culture consist of deep culture (believes, language, norm, rules, knowledge) and surface culture (products, services, art, architecture etc.). Software would be an artifact of surface culture in the context.

Technology

"Technology is knowledge (deep culture) imbedded in products and processes and it is also the products and processes itself" (Kersten, Kersten, Rakowski, 2002, p. 5). As the three researchers suggest there are three different views on technology.

- Instrumental - technology is completely neutral and has no cultural "flavor" at all.
- Substantive - technology is a new culture of its own.
- Critical - technology becomes culture-laden during process of design, development, and use.

The third perspective is the one we choose as a ground view on technology. The two cornerstones of the critical view are technology can be applied for achieving social objectives. Technology is tightly coupled with people and thus their culture.

Culture and technology

The relationship between technology (in the context- software) and culture could be studied in different ways depending on different perspectives on technology and culture. The current trend- that one can develop global software through modification of user interface only- could be explained by application of the instrumental perspective on technology and the reductionist perspective on culture. "The viewpoint of most software designers and engineers can be tracked to both reductionist understanding of culture as language and other symbols and the instrumental theory of technology. In this context software is culturally neutral and can be adapted to every culture through the modification of its user interface." (Kersten, Kersten, Rakowski, 2001, p.6).



Our opinion is that technology and software in particular is not cultural independent. For example the authors of the article showed that three DSS build in three different countries (US, England, and France) were based on different cognitive models and assume a decision maker (the user) to have different attitude to a process of decision making. Another example is Hyperion's discover that the higher grade of flexibility in business approaches is desired in US but not Germany (Coronado, Livermore, 2001, p.23). Kersten says: "Procedures and algorithms to construct and solve problems are implemented in software core. If they are cultural-dependent then the core is also cultural-dependent." (Kersten, Kersten, Rakowski, 2001, p.10).

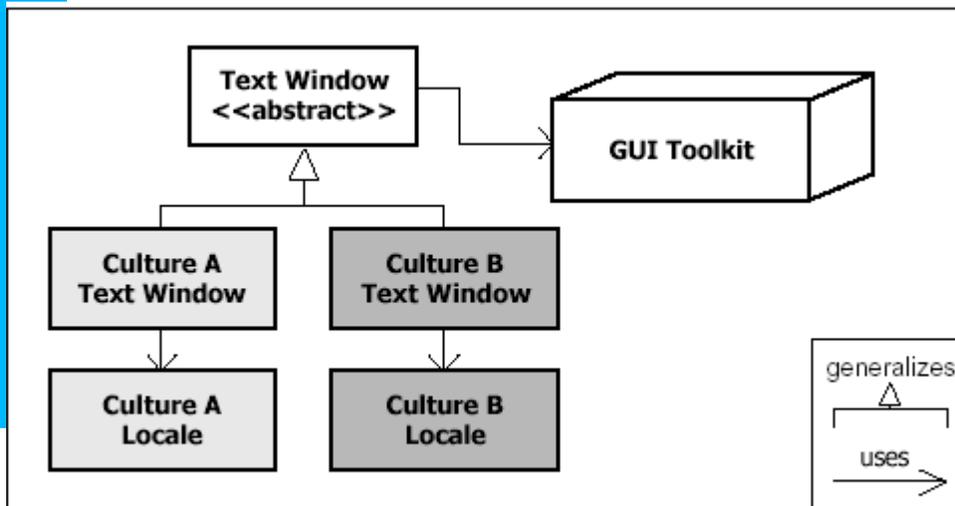
One might wonder if that matters. A user does not have to be concerned with how the application goes about processing information. I'm just interested with the application's interactive elements (GUI) being intuitive and easy to use for me. Dray (2000) has an answer to that kind of reasoning "...if the technology forces user to work in a way based on American practices, but which do not fit their work practices, software is most likely be viewed negatively." (Dray, 2000, p.3). According to Dray understanding of work flow and task structure is very important. This is the exact same thing which Hyperion recognizes and takes care of.

What we are trying to say is that sometimes it does not matter how software core is built. Although, when that core is based on assumptions on, for example business processes to be flexible or strict then the cultural "flavor" (the deep culture) of the core is a matter of a big importance.

Based on the above discussion we suggest a deeper localization, e. g. localization of software core or business logic (originally proposed by Kersten, Kersten and Rakowski, 2001) to have to be considered when development of global software is to be done.

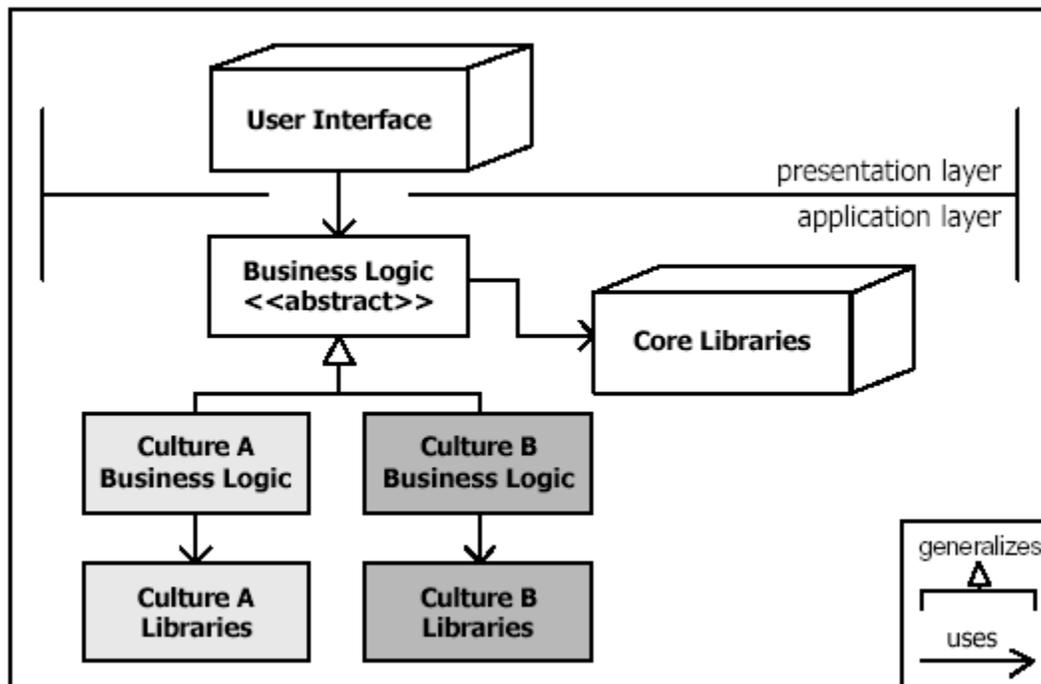
Deeper localization

The following figure represents the current state-of-art of internationalization of software. Localization does not go deeper than the interface layer.



Adapted from Kersten (Kersten, Kersten, Rakowski, 2001, p. 6)

On the picture below one can see that localization includes the “business logic” layer. The two pictures together constitute a good picture of our answer to the research question - if a deeper localization is needed.



Adapted from Kersten (Kersten, Kersten, Rakowski, 2001, p. 12)

6. Discussion

What are the difficulties with producing a methodology for software globalization? Well, the field of human-computer interaction is quite a new science. "For example,

HCI has more evaluation methods than design methods. In addition, HCI substantive knowledge, e.g. guidelines and standards, has no associated guarantee." (Long, 1994, p.190). Besides that many findings that companies make are very often commercially sensitive and as a natural consequence are not published. As one of the researches in the field of software localization wrote to us: " it may be that what they (companies) found out may be commercially sensitive, which leaves people like us (researchers), maybe, reinventing the wheel..." (Yeo_020410). Another difficulty or threat is that we can't actually test our finding on a development of any particular product. The third threat or problem is that we ground our presentation of different methodologies on articles. This material may not be sufficient. The good news is that we contact authors to clear up issues and even to acquire some additional information.

6.1 The choice of Waterfall

Our extended methodology is based on the Waterfall model. Why did we choose it? After all it is rather an old model and one might think no one uses it any more. Well, the model is old and rigid and there are a lot other newer and more flexible models used in the industry today. To name some: *Staged development*, *Evolutionary development*, *Evolutionary Prototyping*, *Modified Waterfall* and *Code and Fix*.

To begin with, we should point out the fact that behind the different choices of which model to use lie differences in objectives, threats, and resource shortage of different projects. According to McConnell (1996) you shouldn't choose *Evolutionary Development* just because some successful software organizations use it. Instead you should choose a model that serves your project best.

Different models focus on achieving different objectives; cost limitation, development time shortening, risk limitation, producing tangible results fast etc. Depending on what objectives are the most important, the same company might sometimes choose different models for different projects. If you have plenty of time, you know the type of application being developed very well, your client does not ask for some of the functionality to be delivered before the whole product is finished then Waterfall may very well be a justified choice for you! If you want to begin coding immediately, the size of the project is small and documentation is not of any particular interest, then you can apply *Code And Fix* model (the most of the labs we've done). In that case there may not be any analysis or/and design stages at all. And of course you can modify Waterfall as you which to have a set of steps and order those steps to serve your project best.

Still as we are considered here with issues of development of global software products it really does not matter which model to pick. The Waterfall model has all the stages the rest of models have (and sometimes more). That why it is a valid choice for using it as a skeleton for our Extended Methodology.

6.2 Web Applications and Software Applications

It seems that one does more addressing the cultural issues (like page layout, structuring presentation of information, icons, metaphors and images) when designing web user interfaces than a GUI for a software application. Probably this feeling originate from so many software applications having a very conventional appearance. What are meant by a conventional appearance are for example square-shaped windows, buttons, menus, etc. Many applications follow look and feel standards set by big software producers like Microsoft or Sun. It seems that currently web development is more sophisticated than software development in trying to address cultural differences of the targeted cultures.

One possible explanation could be that producing sophisticated software interface is a more complex and challenging task then producing a web interface. Besides, web pages use such visual design elements (for example pictures), that aren't used in software GUIs in the same extension. A picture in the context could be an image, closely related to a culture. That could be the famous Brazilian Mountain of Sugar Loaf or a Chinese-styled lotus flower. A Brazilian or a Chinese user would then perceive this cultural marker and feel that the web page or a program is designed for him/her. "Cultural markers have great effects on users' satisfaction to a specific multilingual Web page and can ease users' navigation." (Sun, 2001, p.99). We asked Jose Coronado (manager of the Localization Team at Hyperion) if they recognised and used some kind of cultural variables like those of Hofstede. "This type of concepts is hard to consider and apply in large complex business applications" (Jose_020328) Jose replied. As software applications often have much more complex information processing core, of course development of both GUI and functionality demand more recourse. How elaborate the particular localised GUI is to be is often a question of how much a particular company can invest in the development of that GUI.

6.3 Final thoughts about the Extended Methodology

Our suggestions about design of localized GUIs really should be tested and have at the moment not very much of a practical value. The way we recommend certain things for developers, who target different countries is also somehow obscure. We usually say "when developing for a country with high PD" or "when developing for a country with low UA". The problem is what one should consider high or low. That of course can be decided in a particular situation at hand. One should remember that those values are all relative. When it for example comes to UA dimension then Belgium has high UA comparing to Sweden, because Belgium is ranked 5 while Sweden is ranked 50 by Hofstede.

Besides that Hofstede's research covered as many as 53 countries it is still not the whole world. The possibility of making analogies when targeting a country not in Hofstede's matrix is an open question. The study also involved employees of a single



company (IBM) and it could have influenced the results of Hofstede's research.

When it comes to a theoretical discussion of our suggestions and "discoveries" about design of localized GUIs we feel that all of them could be questioned. The suggestion about structuring menus is controversial as there are some opposite opinions about that. While we recommend making a menu tree deeper so as the user exposed to fewer alternatives at a time, Bruce Tognazzini (famous GUI expert working for Apple Computer, Inc.) advises avoiding elaborate menu trees and using hierarchicals "...only when necessary and only for less-often-used items." (Tognazzini, 1992, p. 314). At the same time a question arise- how many menu items should be considered recommendable for different countries? Suppose we're designing for Belgium (ranked 5- high UA) and Sweden (ranked 50- low UA). Should the menu bar of the Belgium version consist of 3 items (e. g. "File", "Other", and "Help") while the menu bar in the Swedish version's will have 7 items ("File", "Edit", "View", "Insert", "Format", "Tools", "Window", "Help")? The answer to the questions of that character could be acquired through testing and usability evaluations sessions (something we can't afford doing).

The suggestion about how free one should feel introducing new concepts (and even objects or controls) into GUI design depending on the UA value of a target country is questionable as well. There are for example ideas of using "pie" menus when one makes right button mouse click because a "pie" menu is more ergonomic as the usual rectangular menu. More ergonomic means that one has to move around the mouse less when working with pie menus. But users around the globe are used to a pie metaphor being used as a diagram to represent some set of values (for example how a sum of money invested is split between different industries or companies) and not being used as a menu. So how much free exactly one should feel when developing for Belgium and Sweden when it comes to an idea of using these ergonomic pie menus. The answer again is testing and usability evaluations with target users.

The suggestion about development of a set of "feminine" controls is probable the most controversial one. One may agree upon the definition of a femininity used in the thesis but how then do you actually judge what is aesthetic and what is not? It is all about personal opinions and tastes when it comes to colors, images, and aesthetics. However, as we base on other researchers' ideas and articles our suggestions are not produced "out of the thin air".

We have been working with a very good support from Carrie Livermore and Jose Coronado of Hyperion, Alvin Yeo (Malaysia University), Gregory E. Kersten (Concordia University, Montreal, Canada), Mik A. Kersten (Xerox PARC, Palo Alto, USA), and Wojciech M. Racowski (ANS, Bell, Toronto, Canada). Of course we are aware of the fact that even if Carrie and Jose were willing to help they still couldn't give us all the information they had about software development at Hyperion. The good news is that we did have contact with both developers and researchers. That may insure that our paper is up to day and not too theoretical.



We would like to believe that our Extended Methodology is an interesting subject for the target audience to read, criticize, and hopefully use as a source for inspiration and further research.

6.4. Future research

It occurred to us in the late stage of the work that when a mapping between different culture's preferences regarding the GUI design is completed then it would be possible to build tools for development of localized GUIs. To try to make this mapping could be an interesting research goal for another research.

7. Conclusions

Globalization issues in a development process require attention. We believe cultural aspects will clearly have an impact. Developers will achieve greater success and increased profitability through the global distribution and acceptance of their products. As the technology and users of it increases all through the world with new computer-based communication media, data and platforms will lead to an increased demand for global solutions.

Besides the typically linguistic issues there are other issues to deal with like visuals and color, navigation and access and user values and attitudes. When you localize a User Interface Design and try to adapt it to a target country you have to consider different cultural variables to get it accepted. We believe that Hofstede's cultural dimensions will have an impact on this by creating ways of designing menus and controls and structure information in order to avoid uncertainty that match user needs and behaviors in a specific country. The culture variables can also influence such stages as Usability testing in respect of authorities and in the context of Collectivism vs. Individualism.

The relationship between software and culture is based on different perspectives on technology and culture. As a consequence of that we see software as an artifact of surface culture which imbeds deep culture (norms, knowledge etc) of the people who create it. In order to create software for other cultures it is important to know and use the deep culture of the target country. The deeper localization is to assist one in doing it. Designing software interface and software core with globalization in mind one carries out deeper localization and makes global software truly global.



8. LITERATURE

Books

1. Cantú, Marcu, *Mastering Delphi 3*, Sybex, 1997.
2. Del Galdo, Elisa M. & Nielsen, Jakob, *International User interface*, Wiley Computer Publishing, 1996.
3. Esselink, Bert, *A Practical Guide to Localization*, John Benjamins Publishing Company, 2000.
4. Hofstede, Geert, *Cultures and Organizations: Software of the Mind*, New York, McGraw-Hill, 1997.
5. Hofstede, Geert, *Organisationer och kulturer - om interkulturell förståelse*, Lund: Studentlitteratur, 1991.
6. McConnell, Steve, *Rapid development: Taming Wild Software Schedules*, Microsoft Press, 1996
7. Tognazzini, Bruce, *TOG on Interface*, Addison-Wesley Publishing Company Inc, 1991.

Paper in a journal

1. Belge, Matt. *The Next Step in Software Internationalization*, Interactions. January 1995: p. 21-25.
2. Carey, Jane M, *Creating Global Software: a conspectus and review*, Interacting with Computers 1998: p. 449-465.
3. Chen, Benjamin, *Creating a Global Internet Presence: Globalization, Localization, and Translation*, iXL Enterprises, Inc. 1999.
4. Coronado, Jose & Livermore, Carrie, *Going Global with Product Design Process. Does it Make Business Sense?*, Interactions November-December 2001, p. 21-26.
5. Dray, Susan M., *Thoughts from 35000 feet: The evolving real-world context of user-centered design*, Interactions May – June 2000, p. 21-26.
6. Dray, Susan, *Designing for the rest of the world: A consultants observation*, 1996.
7. Kersten, Gregory E., Kersten, Mik A., Rakowski, Wojciech M., *Application Software and Culture: Beyond the Surface of Software Interface*, Internege reports 01, 2001.
8. Marcus, Aaron & Gould, Emelie West, *Crosscurrents: Cultural Dimensions and Global Web User-Interface Design*, Interactions. July - August 2000, p. 32-46
9. Nakakoji, K., *Beyond Language Translation: Crossing the Cultural Divide*, IEEE Software, Vol.13, No. 6, p. 42-46, November, 1996.
10. Nielsen, Jakob, *Designing User Interfaces for International Use*, North-Holland, 1990.
11. Pierce, Todd, *The International Pictograms Standard*, ST Publications, Cincinnati, OH, 218 pp, 1996.
12. Siegel, David A., Dray, Susan M., *New Kid on the Block: Marketing Organisations and Interaction Design*, Interactions March – April 2001, p. 19-24.



Paper in a conference proceeding

1. Khaslavsky, Julie, *Integrating Culture into Interface Design*, CHI 98 1 18-23 April 1998
2. Long, John, *Is HCI Education Getting a Passing Grade from Industry?*, CHI 94, Boston, Mass, 24-28 April 1994.
3. Marcus, Aaron, *Globalization of User-Interface Design for the Web*, Proceedings of the 5th Human Factors and the Web conference 3 June 1999 in Gaithersburg, MD
4. Shaoyi, Ey, *Interplay of Language and Culture in Global E-commerce: A Comparison of Five Companies' Multilingual Websites*, SIGDOC '01, October 21-24, 2001, Santa Fe, New Mexico, USA.
5. Sun, Huatong, *Building A Culturally-Competent Corporate Web Site: An Exploratory Study of Cultural Markers In Multilingual Web Design*, SIGDOC '01, October 21-24, 2001, Santa Fe, New Mexico, USA
6. Yeo, Alvin W., *Global-Software Development Lifecycle: An Exploratory Study*, 2001, SIGCHI '01, March 31-April 4, 2001, Seattle, WA, U.S.A.

Web sites

1. Localization Industry Standard Association (LISA), <http://www.lisa.org>
2. LISA definitions, <http://www.lisa.org/info/faqs.html>
3. Brooke, John, *SUS - A quick and dirty usability scale*, Redhatch Consulting Ltd., <http://www.usability.serco.com/trump/documents/Suschapt.doc>

Mail

Jose Coronado, Carrie Livermore at Hyperion:

Livermore_020328
Livermore_020416
Livermore_020423
Livermore_020425
Livermore_020507
Coronado_020328
Coronado_020416

Alvin. W. Yeo:

Yeo_020402
Yeo_020410

Gregory E. Kersten, Mik A. Kersten, Wojciech M. Rakowski

Kersten_020424
Rakowski_020501
Kersten_020501



9. DEFINITIONS OF TERMS

Globalization (G11n)

"Globalization addresses the business issues associated with taking a product global. In the globalization of high-tech products this involves integrating localization throughout a company, after proper internationalization and product design, as well as marketing, sales, and support in the world market." (LISA)

Internationalization (I18n)

"Internationalization is the process of generalizing a product so that it can handle multiple languages and cultural conventions without the need for re-design, Internationalization takes place at the level of program design and document development." (LISA)

In particular, internationalization involves:

- Ensuring that all localizable elements are extracted from the source code (e.g. by collecting all localizable items in external resource files);
- Ensuring that the design of the user interface (UI) is flexible and neutral;
- Ensuring that the relevant character set is supported;
- Ensuring that regional standards are supported (see Data formatting)
- For Web sites, ensuring that text embedded in graphics is easily localizable.

Localization (L10n)

"Localization involves "taking a product and making it linguistically and culturally appropriate to the target locale (country/region and language) where it will be used and sold". (LISA)

Levels of localization [7]

Less risk, less return

Translate nothing

Translate documentation and packaging only

Enable code

Translate software menus and dialogs

Translate online help, tutorials, and sample and readme files

Add support for locale-specific hardware

Customize features for locale (full localization)

More risk, more return

Adapted from Carey, 1998, p. 8

Full localization includes "the entire user interface including menus, dialogs, title bars, status messages, error messages, icons, bitmaps, on-line help, computer-based training materials, documentation, sample programs or files, and a translated macro language." (Carey, 1998, p. 8)

According to Carey (1998) full localization has the benefits of addressing a broader

audience, making the software more accessible, easier to use, better accepted, easier to support. The end result for full localization allows for more sales and the ability to charge more. The author points out that under some conditions it may be reasonable to localize partially. "Some markets in various countries, e.g. Sweden may tolerate an English version of the product. The Japanese will not nor the French." (Carey, 1998, p. 8) However, if the product is targeted to the end-user rather than the developer market, then it according to Carey should be fully localized.

Carey accentuate that the decision of how to localize is of course a monetary one and depends on the level of investment the company is willing to make.

Translation

"Translation is the process of converting written text or spoken words to another language. It requires that the full meaning of the source material be accurately rendered into the target language, with special attention paid to cultural nuance and style." (Esselink, 2000)

Deep culture

Deep culture consist of believes, language, norm, rules, knowledge. (Kersten, Kersten, Rakowski, 2001)

Surface culture

Surface culture could be described in terms of products, services, art, architecture etc. (Kersten, Kersten, Rakowski, 2001)

GUI control

"A control is technically a predefined window with a specific behavior, some properties, and some methods." (Cantù 1997). An example of a GUI control could be a button, a scroll list or a form.

10. Appendix

Appendix 1: Table from: Hofstede, Geert, *Organisationer och kulturer - om interkulturell förståelse, 1991.*

PDI: Power distance index

IDV: Individualism index

MAS: Masculinity index

UAI: Uncertainty avoidance index

LTO: Long-term orientation index

	PDI		IDV		MAS		UAI		LTO	
	rank	score	rank	score	rank	score	rank	score	rank	score
Arab Countries	7	80	26/27	38	23	53	27	68		
Argentina	35/36	49	22/23	46	20/21	56	10/15	86		
Australia	41	36	2	90	16	61	37	51	15	31
Austria	53	11	18	55	2	79	24/25	70		
Bangladesh									11	40
Belgium	20	65	8	75	22	54	5/6	94		
Brazil	14	69	26/27	38	27	49	21/22	76	6	65
Canada	39	39	4/5	80	24	52	41/42	48	20	23
Chile	24/25	63	38	23	46	28	10/15	86		
China									1	118
Columbia	17	67	49	13	11/12	64	20	80		
Costa Rica	42/44	35	46	15	48/49	21	10/15	86		
Denmark	51	18	9	74	50	16	51	23		
East Africa	21/23	64	33/35	27	39	41	36	52		
Ecuador	8/9	78	52	8	13/14	63	28	67		
Finland	46	33	17	63	47	26	31/32	59		
France	15/16	68	10/11	71	35/36	43	10/15	86		
Germany FR	42/44	35	15	67	9/10	66	29	65	14	31
Great Britain	42/44	35	3	89	9/10	66	47/48	35	18	25
Greece	27/28	60	30	35	18/19	57	1	112		
Guatemala	2/3	95	53	6	43	37	3	101		
Hong Kong	15/16	68	37	25	18/19	57	49/50	29	2	96
India	10/11	77	21	48	20/21	56	45	40	7	61
Indonesia	8/9	78	47/48	14	30/31	46	41/42	48		



Iran	29/30	58	24	41	35/36	43	31/32	59		
Ireland (Rep of)	49	28	12	70	7/8	68	47/48	35		
Israel	52	13	19	54	29	47	19	81		
Italy	34	50	7	76	4/5	70	23	75		
Jamaica	37	45	25	39	7/8	68	52	13		
Japan	33	54	22/23	46	1	95	7	92	4	80
Malaysia	1	104	36	26	25/26	50	46	36		
Mexico	5/6	81	32	30	6	69	18	82		
Netherlands	40	38	4/5	80	51	14	35	53	10	44
New Zealand	50	22	6	79	17	58	39/40	49	16	30
Nigeria									22	16
Norway	47/48	31	13	69	52	8	38	50		
Pakistan	32	55	47/48	14	25/26	50	24/25	70	23	0
Panama	2/3	95	51	11	34	44	10/15	86		
Peru	21/23	64	45	16	37/38	42	9	87		
Philippines	4	94	31	32	11/12	64	44	44	21	19
Poland									13	32
Portugal	24/25	63	33/35	27	45	31	2	104		
Salvador	18/19	66	42	19	40	40	5/6	94		
Singapore	13	74	39/41	20	28	48	53	8	9	48
South Africa	35/36	49	16	65	13/14	63	39/40	49		
South Korea	27/28	60	43	18	41	39	16/17	85	5	75
Spain	31	57	20	51	37/38	42	10/15	86		
Sweden	47/48	31	10/11	71	53	5	49/50	29	12	33
Switzerland	45	34	14	68	4/5	70	33	58		
Taiwan	29/30	58	44	17	32/33	45	26	69	3	87
Thailand	21/23	64	39/41	20	44	34	30	64	8	56
Turkey	18/19	66	28	37	32/3	45	16/17	85		
Uruguay	26	61	29	36	42	38	4	100		
USA	38	40	1	91	15	62	43	46	17	29
Venezuela	5/6	81	50	12	3	73	21/22	76		
West Africa	10/11	77	39/41	20	30/31	46	34	54		
Yugoslavia	12	76	33/35	27	48/49	21	8	88		
Zimbabwe									19	25