Revista Română de Interacţiune Om-Calculator 7 (2) 2014, 161-194 © MatrixRom

Towards a Culture-Adaptable User-Interface Architecture

Iyad Khaddam, Jean Vanderdonckt

Université catholique de Louvain – Lilab
Place des Doyens 1, Louvain-la-Neuve, Belgium
E-mail: {iyad.khaddam, jean.vanderdonckt}@uclouvain.be

Abstract. Culture plays a role in the way humans interact with the machine. It implies changes on the UI to fit cultural needs, practices and preferences. Knowledge on these implications is sparse and insufficient until the moment. Employing culture knowledge as rules in the design and development of software user interfaces looks far to achieve. Even though, user interface designers and developers still need to consider differences in user interfaces. Our approach to support user interface developers and designers is to propose a user interface architecture that enhances their ability to apply changes imposed by culture. Our contribution is to analyze culture studies to identify concerns and requirements for such architecture, propose an architecture that satisfies some of these requirements and discuss an idea towards an enhanced one.

Keywords: Culture, UI architecture, HCI, culture models.

1. Introduction

End users primarily interact with information systems differently depending on cultural and personal differences. Addressing cultural differences of use in designing User Interfaces (UIs) may improve usability and acceptance of these UIs and their corresponding information systems. Considering these differences in the UI development lifecycle might be essential for a usable UI (i.e. language).

Supporting culture in UI is a challenge to go global. The first part of this challenge is to know “what is the difference”, while the other part is “how to support”. Culture knowledge (the “what is the difference”) comes from culture studies explaining differences among cultures practices and values. Well known culture aspects in practice are language, reading and writing direction, fonts, measures, calendar and date and time formats. These aspects are known as explicit culture implications on UI and are essential in the acceptance of software by users (Evers and Day, 1997). Software industry transformed many of these aspects into explicit knowledge using

UI adaptation rules and libraries of code. Java provides an internationalization checklist of culturally-dependent data and tutorials on building international software (http://docs.oracle.com/javase/tutorial/i18n/ intro/checklist.html). It also provides culture-adapted libraries like Calendar class (http://docs.oracle.com/javase/1.4.2/docs/ api/java/util/Calendar.html).

Culture implications on UI are not only limited to explicit culture. Researchers have been studying these implications for over a decade aiming to acquire guidelines and rules from culture studies to help UI designers better satisfy end-users culture needs as part of enhancing usability of UI. The generic process to elicit UI culture guidelines involves cross-domain knowledge and expertise. It starts with culture experts (anthropologists, psychologists and others) who conduct their research on culture observations to create culture patterns. Promising culture patterns then are examined by Human Computer Interaction (HCI) experts who in turn try to find signs of implications on the domain. When a sign is found, HCI researchers may look for evidences in real world, or run experiments to prove the applicability of this culture study in UI. Following either approach, HCI researchers represent their findings through culture guidelines. This process is depicted in Figure 1.

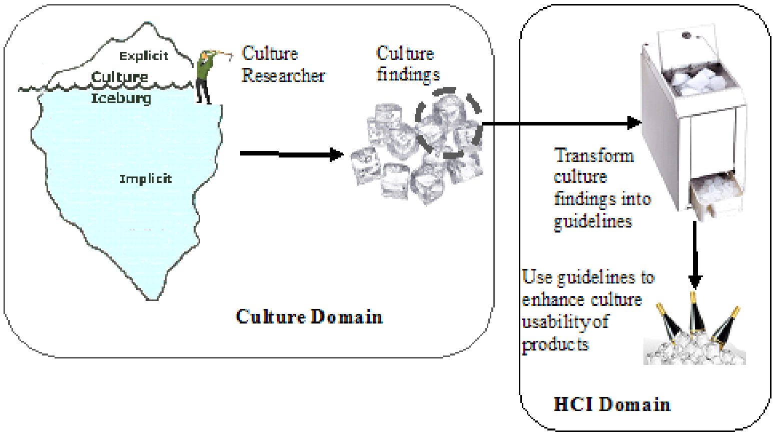

Explicit
Culture
Iceburg
Implicit
Culture Researcher
Culture findings
Culture Domain
Transform culture findings into guidelines
Use guidelines to enhance culture usability of products
HCI Domain

Figure 1. The generic process to acquire culture guidelines

Even with sufficient knowledge on implicit culture implications, validity and applicability of this knowledge might be questioned. It might not be easy to everyone to accept that Arabs prefer green color in university web sites, as explained by Marcus in (Marcus, 2009). One may also argue that this knowledge may not be extensible to other domains, like e-commerce web-sites, which limits applicability. In this article, we do not discuss the validity or applicability of these culture findings. One can find a reason or another to doubt implicit culture knowledge, so they can't be considered as absolute facts. We believe that supporting implicit culture in UI is a creative process that is directed by culture knowledge. The continuous need for customization, modification and adaptation is part in our mindset.

Our knowledge on culture is not complete and what we know is just the top of the iceberg. Supporting culture in UI development based on domain-dependent, incomplete and doubted knowledge results in limited use of such solutions. Dealing with implicit culture knowledge as usability rules that control the UI design can be found in the works like in (Reinecke, 2010). We follow a different approach that is supported by culture literature review: with current level of knowledge on culture, user's cultural preferences are unpredictable. Based on that, we believe that a practical approach to support culture development is by enhancing adaptability of UI to culture changes. Enhancements address UI architecture to enable UI developer to implement culture changes on the UI. A UI developer denotes UI designers, programmer, UI usability experts and other roles may be involved in developing a UI. A culture-adaptable UI (CA-UI) architecture is an architecture that enhances UI developers' ability to implement change requirements imposed by culture.

To explain our approach, we give an example from movies industry, which is full of culture signs. An American movie may need to be translated into French to be broadcasted in France on a popular scale. Several possible solutions for this culture adaptation: re-produce the movie with actors speaking in French, which is a very costly and a non-practical solution, or depending on the technology used in the movies to reproduce only the sound track, which is less costly and more effective as the targeted culture client can re-produce the sound tracks using native actors who are able to convey the movie messages in their appropriate culture way. A third style, also depend on the underlying technology, is add translation text on the screen. It is obvious from this example that film technology enables movie producers to implement some of their creative solutions to support culture in their products. Anyway, not everything can be changed. For instance, the city of the famous cartoon character "Sponge Bob" is called bikini bottom.

Arabs in general are conservative people and won't feel comfortable to hear the word “bikini bottom” word in a TV cartoon series for kids. The Arabic version of this series is dubbed and the name is changed into “The Hamour cliff”, name that may convey part of the original message (deep under the sea) but not the funny part of it. The problem in the final product is that the original name of the city appears in some scenes on the screen, and kids know English from school (Figure 2). If the movie technology allows modification on the movie frame level at reasonable costs, the final version could support the Arabic culture better by changing the sign text into Arabic in figure 2.

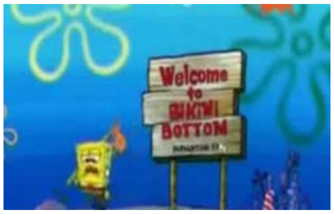


Figure 2. Technology barrier to appropriate adaptation to Arabic culture

The first style in the example above (reproducing the movie) is similar to localization of UI software, an approach developers may prefer to escape due to high efforts for maintenance and future updates. Providing the UI developer with appropriate tools may lead to avoiding such an approach.

To reach our goal in searching for CA-UI architecture, we need to first review literature on culture to identify challenges and learn more about the nature of culture changes and their scope. Section 2 presents culture domain. Section 2 is fairly long to give the reader a broad view of culture domain and implications on UI. Knowledgeable readers in culture domain and HCI may skip sections 2.1 and/or 2.2. Section 2 motivates and justifies our approach. It lists also our concerns and requirements; our first contribution in this article. As our approach to address culture is based on changes imposed by culture, we use the reference framework for analyzing changes proposed by Filieri et al in (Filieri, 2012). This reference framework helps to establish a global view of our CA-UI architecture. We present this framework and our proposed architecture in section three. The proposed architecture partially satisfies requirements stated in section two. Requirements that are not satisfied are tackled in the discussion section (section four) by discussing difficulties to satisfy and introducing ideas and approaches. Section five is dedicated for conclusion and future work.

2. Culture

This section aims to present some researches in the domain of culture, and explain how this knowledge is presented and used in the HCI. It also tries to address culture studies from the point of view of software development, by trying to identify culture groups and scope of culture studies as addressed by HCI researches.

2.1 Culture models and dimensions

Culture models aim to identify culture dimensions and classify groups of people according to these dimensions. Many anthropologists and scientists conducted research on culture, trying to elicit dimensions that play important role in this domain, resulting in a lot of dimensions that overlap sometimes.

Many culture models exist like the model (with six dimensions) proposed by Nancy J. Adler (Adler, 2002) or the model of Florence Rockwood Kluckhohn and Fred Strodtbeck (Kluckhohn & Strodtbeck, 1961) (they identified five dimensions) and others. For conciseness, we present only two models to give the reader an idea of the modeling of culture approach.

Edward T. Hall, an anthropologist, distinguished cultures on the way of communicating from “high-context” to “low-context”. A high-context communication is one in which little has to be said or written because most of the information is either in the physical environment or within the person, while very little is in the coded, explicit part of the message. He described various values in his book "Beyond Culture" (Hall, 1989): (1) Context, (2) Polymorphic or Monochromic Time, (4) Preferred Message Speed and (5) Space.

The culture model of Greet Hofstede, a Dutch cultural anthropologist, is the one that got a lot of interest in the HCI domain. Maybe because it covers the largest number of countries (national level) and also quantifies cultures by providing measures for each country in the study on each dimension of his model. He conducted a research on cultural differences between the years 1978-1983. He used statistical analysis on IBM workers in 53 countries and formulated his theory that world cultures vary along consistent, fundamental dimensions. He published his research in Cultures and Organizations: Software of the mind (Hofstede, 1997). The main characteristic of Hofstede's culture model is that it is more value-oriented than about cognition and perception differences. He classifies cultural differences into two layers: Practices (the upper layer) and Values (the core layer). Practices are Symbols: words, pictures, gestures or objects that carry a particular meaning (words and jargon, dress, flags), Heroes: are persons, alive, dead or imaginary who possess highly prized characteristics thus serve as models for behavior (Snoopy in USA, Asterix in France) and Rituals: collective activities that are considered essential (ways of greeting, paying respect to others, social and religious ceremonies). Values are broad tendencies to prefer certain states of affairs over others. Values are feeling with an arrow to it: a plus and a minus side (evil vs. good, dirty vs. clean, ugly vs. beautiful, unnatural vs. natural, abnormal vs. normal, irrational vs. rational).

Hofstede identified five dimensions: (1) Power Distance: the extent to which less powerful members expect and accept unequal power distribution, (2) Collectivism vs. individualism: Individualism in that everyone is expected to look after one's self or immediate family only. Collectivism implies that people are integrated from birth into strong, cohesive groups that protect them in exchange for unquestionable loyalty, (3) Femininity vs. masculinity: traditional assignment to masculine roles of assertiveness, competition, and toughness in contrast to feminine roles of orientation to home and children, people, and tenderness. Different professions in different cultures are dominated by different genders, (4) Uncertainty Avoidance: the extent that they feel anxiety about uncertain or unknown matters, as opposed to the more universal feeling of fear caused by known or understood threats. Cultures vary in their avoidance of uncertainty, creating different rituals and having different values regarding formality, punctuality, legal-religious-social requirements, and tolerance for ambiguity, (5) Long- vs. short-term orientation: Asian countries are oriented to practice and the search for virtuous behavior while Western countries are oriented to belief and the search for truth

Researches in cognitive and perception differences among cultures are also interesting to HCI researchers. Nisbett was interested in studying cognition and perception differences among cultures. With colleges, his studies led to the following cognition orientation differences (Nisbett & Masuda, 2003) that appear to be particularly magnified when Asian and western cultures are compared. We present a sub-set of their work that we see related to HCI:

1- **Field independence vs. field dependence:** East Asians are inclined to focus their attention broadly on the field, and find it more difficult to make a separation between an object and the field than do Westerners.

   In the rod and frame experiment (figure 3), the participants were asked to tell if the rod was vertical or not (rod and frame are independent from each other). Chinese where highly influenced with the position of the frame. This indicates that Chinese are field dependent in their perception to objects.

1- **Attention to object vs. field:** East Asians attend to objects in relation to the field, and will be more thrown off by a presentation of the object against a novel field than would Westerners. Westerners are inclined to attend to focal objects.

   In an experiment, Nisbett presented a video to Japanese and American participants. Japanese mentioned information about the field 65% more than Americans did. In another test, the same groups were presented with the images like in Figure 4 which has the same object but in different fields (different background). Japanese made more errors guessing the object was presented before.

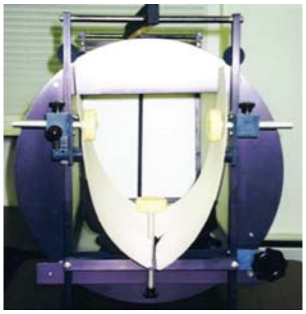


Figure 3 Nisbett's Rod and frame test apparatus

2- **Causal Attribution, prediction and postdiction:** East Asians are inclined to explain events by reference to interactions between the object and the field, while Westerners are inclined to explain events by reference to properties of the object With regard to postdiction (such as post hoc explanation), East Asians are not as surprised by unanticipated outcomes as Westerners.

3- **Categorization:** East Asians classify objects and events on the basis of relationships and family resemblance, whereas Westerners classify them on the basis of rule-based category membership. In an experiment, children from China and America where given three images (cow, chicken and grass). Americans put the cow and the chicken in one group based on the rule that they are both animals. Chinese put cow and grass in one group based on the family relation between them.
For HCI, designers need to develop a user-friendly navigation system that would help him find what he look for easily. Researchers discussed designing culture-aware menus based on culture categorization differences (Nawaz & Clemmensen, 2007).

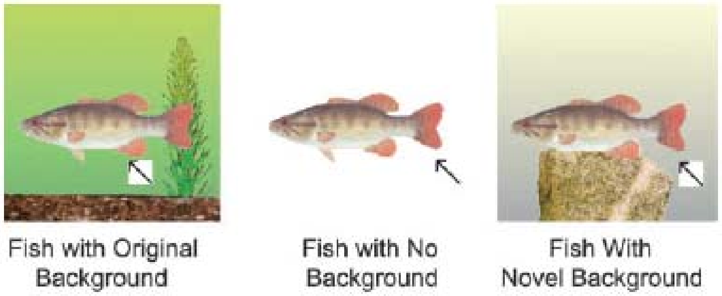


Figure 4 Nisbett's test on field dependence/independence

2.2 Culture knowledge in HCI

Based on culture models and cognitive and perception differences among cultures researches, HCI researchers tried to transfer culture knowledge into their domain. The result is represented as a set of culture guidelines to provide UI designers with recommendations on how to build cross-culture designs. This process is followed for both types of culture studies: cultures models and studies on cognitive and perceptual differences among cultures.

Aaron Marcus tried to study Hofstede's culture dimensions implications on HCI. In his research, he focused on finding evidences that relate Hofstede's national culture dimensions with Web design (Marcus & West

Gould, 2000). For each dimension, he explained the affected aspects of User Interface Design for the Web. Other researchers conducted similar researches like in (Ackerman, 2002) and in (Dormann & Chisalita, 2002). Table 1 shows a sample of their results for the Power distance dimension in Hofstede's model.

In the process of looking for evidences of Hofstede's dimension, researchers mapped UI characteristics to these dimensions. Robbins and Stylianou (Robbins & & Stylianou, 2002) studied commercial web sites in several geographical regions. They mapped power distance with the presence of organizational charts, bio-sketches of top leaders, and messages from CEOs. Uncertainty avoidance is mapped to listings of job openings, cookie disclosures, etc. Site registration requirements, security provisions, and privacy policy statements would denote individualism/collectivism and annual reports and financial highlights represent masculinity/femininity on the site. Long/short orientation is mapped to the presence or lack of a search engine, site map, FAQ, corporate history, etc. Rajkumar (Rajkumar, 2003) examined Indian and U.S. university websites. They interpreted individualism throughout by way of frequent pictures of individuals, direct address (using "you" as opposed to "we"), personalization features, expression of private opinion, individual success stories, while collectivism is reflected by displaying images of groups, use of formal speech, including mission statements that impact the larger group, and stating opinions on group behavior.

Table 1. Implications of Power Distance dimension on web design

UI design aspect	**High Power Distance**	**Low Power Distance**
Access to Information	Highly	Less highly
Hierarchies in mental models	Tall	Sallow
Emphasis on the social and moral order (nationalism or religion) and it's symbols	Significant/frequent	Minor/infrequent use
Focus on expertise, authority, experts, certifications, official stamps or logos	Strong	Weak
Prominence given to leaders vs. Citizens, customers or employees	High	Low
Importance of security and restrictions or barriers to access	Explicit, enforced, frequent restrictions on	Transparent, integrated implicit freedom to roam

	users	
Social roles used to organize information (e.g. a manager's section is obvious to all but sealed off from non-managers)	Frequent	Infrequent

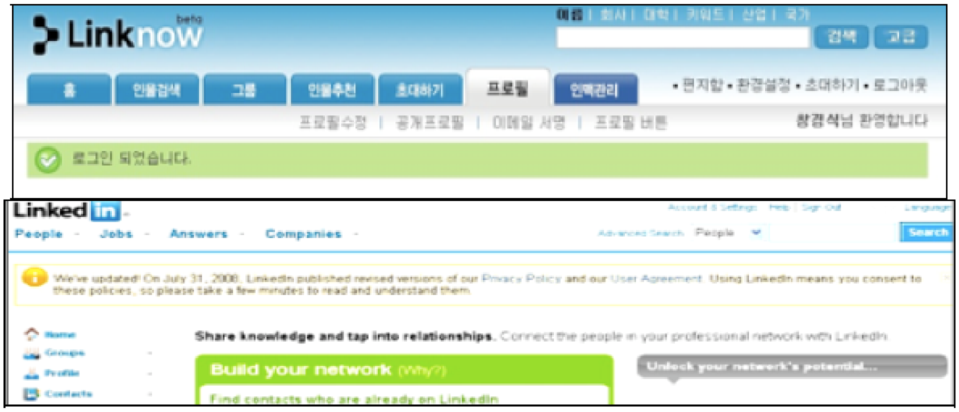
Mapping web site characteristics to culture dimensions is interesting, but incorporating these findings as rules in design needs further research to be more valid. For the current time, we see them useful to UI designers to have clues on the implication of culture on web design.

Callahan (Callahan, 2005) examined university web sites from different countries. He concludes that "Hofstede's model may be helpful in interpreting cultural similarities and differences in graphical design". Some of his findings are: white, and sometimes purple or blue, background was commonly chosen by Malaysian (low power distance country), vivid colors are dominant and the number of colors on the page is large, and also make heavy use of animation. Sweden (Feminine country) had a stronger preference for white backgrounds. Colors were used carefully as a background for links. Blue and yellow, the Swedish national colors are a frequent theme, in addition to grey, purple, pink and dark red. Although Callahan’s results are interesting, but also further validity tests is needed. Besides, his findings are coming from university web sites and generalizing these findings might be questioned.

Marcus and Krishnamurthi examined the user interface components of social networking sites (SNS) in three countries: Korea, Japan and United States (Marcus & Krishnamurthi, 2009). They observed several differences that they relate to culture differences. They argue that "People you may know" feature in Facebook (USA, low Uncertainty avoidance) creates uncertainty as one begins to wonder if they actually know the member mentioned there. Authors don't know of any Japanese or Korean Social Network Services provider who implemented such feature. They also observed a symmetrical design for Korean SNS (high Uncertainty avoidance) and asymmetrical design for USA SNS (low UA) as in figure 5.

Implications of culture cognition and perception on HCI are also not fully understood yet. HCI researchers are still conducting experiments searching for new relations or verifying known ones. Rau and colleges summaries a comprehensive cross-culture design studies for HCI in (Rau, Plocher, & Chong, 2011) and created a list of guidelines. Thanks to their work, we reached examples we present here.

Inspired by Nisbett's work (categorization), Choong conducted an experiment on a web department store to know if information organization is consistent with their natural way of organizing objects. The users were Chinese and Americans and he found that different cultures focus on different attributes of objects. Americans tend to focus on functional attributes while Chinese focus on thematic ones. Choong experiment showed that culture-aware organization of information affects performance of users. For example: Americans prefer to see products in a department store organized according to function: cleaning supplies, linens and furniture. Chinese prefer to see them organized by themes (in this case, the different rooms of a house): kitchen, bathroom and bedroom.


Linknow beta
이름 | 회사 | 대학 | 키워드 | 산업 | 국가
검색
고급
홈
인물검색
그룹
인물추천
초대하기
프로필
인맥관리
• 편지함 • 환경설정 • 초대하기 • 로그아웃
프로필수정 | 공개프로필 | 이메일 서명 | 프로필 버튼
황경식님 환영합니다
Linked in
People
Jobs
Answers
Companies
Account & Settings
Help
Sign Out
Language
Advanced Search
People
Search
Home
Groups
Profile
Contacts
Build your network (Why?)
Unlock your network's potential...
Find contacts who are already on LinkedIn

로그인 되었습니다.

We've updated! On July 31, 2008, LinkedIn published revised versions of our Privacy Policy and our User Agreement. Using LinkedIn means you consent to these policies, so please take a few minutes to read and understand them.

Share knowledge and tap into relationships. Connect the people in your professional network with LinkedIn

Figure 5. Symmetrical (Linknow, South Korea) vs. asymmetrical (LinkedIn, USA) design

Rau and Liang (2003) conducted a survey to classify web users as high context or low context according to Hall's communication style dimension (Hall, 1989). They conducted experiments on those users and found that high context users browse information faster and required fewer links to find information than did low context users. High context users where inclined to get lost and disoriented in hypertext. Low context users were less inclined to get lost. A well designed navigation support is recommended for users from high context cultures.

Different cultures have different attitudes towards time. monochronic (Time is divided into segments that can be easily scheduled and "spent.")

and polychronic (perceive time in a less rigid, more flexible way). Zhao experimented alternative ways of structuring information to measured speed and performance of participants on search and retrieval tasks (Zhao , 2002). Monochromic users performed significantly better on tree hierarchy of information while polychromic users performed significantly faster on network hierarchy.

Colors preference by cultures was also a field of interest for researchers. To create a complete list of culture preference for colors, a huge research effort is needed to cover diversity of cultures. Prabhu and Harel (Prabhu, 1999) studied user's preference of colors and graphical design for digital imaging products in China and Japan. They found that Japanese men prefer single color fonts, simple, while Japanese women, Chinese men and women preferred multiple colors, highlighted or emphasized fonts. Japanese preferred pastel colors for both welcome screens and interaction screens. Chinese men preferred Chinese colors, while women preference was mixed between Chinese and pastel colors.

Minocha, French, and Smith (Minocha , 2002) offered a study for e-Finance web sites in India and Taiwan. Red for Indian users is associated with vitality prosperity and health. Red is considered stimulating and shows ambition and initiative. Saffron is considered auspicious for Hindus, Sikh, Jains and Buddhists. The combination of red and saffron can be considered to signify prosperity and growth for current and prospective customers. Language and localized graphical design as cultural issues for Taiwanese e-Finance web sites. A fish conveys the message of prosperity. The use of cartoon iconography forms part of the Asia-Pacific cultural phenomena.

Metaphors are culture-dependent. Initial users of the Mac in United Kingdom perceived the trashcan as the mailbox. The "Home" page denotes the starting page in web design. The graphical representation can vary across cultures. Everyday object can be used in design, but they may look differently across the world, like paper sizes, telephones, office supplies, and signs. Cultural symbols, heroes and rituals are also cultural. UI designers should be aware of then to prevent misunderstanding or even insulting users unexpectedly. Hand gestures are one example of potential problems. The same hand gesture can mean be interpreted differently or oppositely for different cultures. The "thumb up" means in the U.S "good" and this can be used in graphical design. The same gesture is an insult in Australia.

SNS is also addressed by researchers on cognitive and perceptual differences. Fogg and Lizawa (Fogg , 2008) compared the persuasiveness of Facebook from the United States and Mixi from Japan. Their observations showed that Facebook's persuasive design is more assertive and mechanistic, while Mixi's approach is subtle and indirect. They argued that this is due to cultural differences. Lewis and George (Lewis, 2008) surveyed deception behavior in SNS. Koreans tend to provide incorrect information about their job, salary and physical appearance, while Americans tend to not be truthful about where they lived, age and interests.

The findings of HCI researchers are accumulated as culture guidelines for UI designers. To give an idea, we present some of the guidelines gathered by Callahan in table 2, and part of guidelines gathered by Rau in table 3.

Table 2. A subset of guidelines proposed by Callahan

Dimension	**High**	**Low**
Power distance	symmetry tall hierarchies focus on official seal, national emblems photographs of leaders monumental buildings	asymmetry shallow hierarchies photos of students rather than faculty images of both genders images of public spaces images of everyday activities
Uncertainty Avoidance	limited choices restricted amounts of data limited scrolling references to daily life redundancy	variety of choices long pages with scrolling abstract images

Table 3. A subset of guidelines proposed by Rau et al

Category	**Guidelines**	**Dimension**
G1 Language and format	G1.1 Use unambiguous language	Cognitive
	G1.3 Accommodate text reproduction methods	Perceptual
	G1.5 Use an appropriate method of sequence and order in lists	Cognitive
	G1.6 Linguistic differences	Cognitive

	G1.8 Text directionality	Cognitive and perceptual
	G1.11 Be aware that variations exist within the same language	Cognitive

2.3 Importance of culture in HCI

The importance of supporting different culture aspects varies according to the nature of this aspect. Explicit culture is essential to the acceptance of UI by the end user (Evers and Day, 1997). It contains essential requirements that affect usability like language.

On the other hand, it is difficult to determine the importance of implicit culture aspects to the user. Implicit culture plays as a differentiating property of the final software product. For example: would a user belonging to a culture group use an application with a color theme not in the list of his culture group preference? The answer is probably yes if he doesn’t have an alternative. Innovative products then may not need to consider these aspects unless market factors are pressing. Android operating system didn’t have support for Arabic language in early versions, but that didn’t prevent Arabic users from buying and using Android mobiles or even develop applications for Arabic language (using images to display text). Notice that Arabic is an explicit culture aspect and still the product was not highly affected.

Some implicit aspects are more critical: Would a user who belongs to a religious culture group use an application (i.e. a game application) that contains offensive symbols to his religion? The answer is probably no especially if he is a faithful follower.

From the market point of view, importance of supporting culture can be simply evaluated as: if opponents provide a culture property, we should follow. If the product is targeting a specific culture, we should adapt to that culture. For UI developers, the above point of view can be interpreted as: we should be prepared for changes.

2.4 Culture groups

Culture groups are groups of people sharing the same culture. We consider culture groups as the steps on the axis of change to software UI. When the UI developer identifies a new culture group, a set of changes may be applied on the software UI to meet this specific group’s UI requirements.

Identification of these culture groups is important to relate appropriate changes (UI requirements) to each group.

If we follow culture studies based on models, people can be grouped based on culture dimensions. Cognitive and perceptual differences among culture may provide diverse criteria to create groups of people. The hard part in identifying universal culture groups is that culture dimensions my overlap and so do criteria in perceptual and cognitive studies. We didn't find a research to unify these dimensions or analyze the inter-relation between them. For instance: Are monochronic cultures related to Power Distance dimensions? If there is no relation, guidelines for monochronic cultures are not applicable for cultures with high power distance? If a relation exists, culture findings and guidelines should be analyzed further should to determine the applicability for each resulting sub group.

Hofstede explains that people belong to different groups and categories at the same time, so people unavoidably carry several layers of mental programming corresponding to different levels of culture (Hofstede , 1997). He identifies the following levels: national, regional, ethnic, religious, gender, generation and/or social class level. These levels can form criteria to identify groups of people with shared cultural practices and values. Even if these groups are established, ambiguity might be found as the same person can belong to different groups at the same time thus his current mental program is hard to identify clearly and easily. Take for example a Belgian woman (gender) who lives in Mons (Wallonia region), speaks French (language), and believes in Islam (religion). The mental program is based on several culture levels. What are the implications of each level on HCI? What is the cross-relation among these culture groups? Which prevails at a certain moment or situation?

Hall's description of culture as "culture is communication" adds more complexity to culture groups identification. He relates culture to communication; the same person can behave then differently according to the communication context. Culture groups are then dynamic groups and no more static. An example on the same Belgian woman mentioned above: if she is with a Muslim friend in a restaurant, both are not in favor of ordering Belgian beer (alcohol is prohibited in Islam). If she invited a non-Muslim friend (sharing the same other culture levels), we are not sure how tolerant she would be to offer him a Belgian beer (a preferred drink for Belgians). In

HCI, a restaurant self-service application that allows the clients to order while trying to adapt to client's culture is challenged to successfully apply the right cultural rules for the right person in the right communication context (should the application show beer offers to that woman or not?).

Culture groups identify the targeted users of the User Interface. This step is subtle to build a usable user interface and apply the right culture guidelines. In practice, developers identify a reduced set of users groups (probably based on national level) and take the charge of updating the software to support other groups or sub groups later. It seems this approach will keep going on considering limitations on culture knowledge at this time, not to mention its practicality.

To be pragmatic, we use a broad definition for culture groups: a set of users who share the same practices, preferences, attitudes, behaviors and/or values in the domain of interest. This definition accepts groups like users of a mobile phone, because they share common practices.

2.5 Scope of culture implications on UI

As culture groups represent the steps on the axis of changes, we need to determine what changes are related to culture and what are not. Providing an answer to this question may help in provisioning types of changes that might be related to each culture group.

Cameleon Reference framework defines adaptation as a trilogy of: Framework, Environment and User (Calvary, 2003). One might think that culture is related to adaptation to user only. Literature shows a broader view of culture implications. Culture affects not only all three dimensions of adaptation defined in Cameleon reference framework, but the software development in general. If adapting UI refers to modifying an existing application UI, adaptation to culture affects building the software itself. In this sense, adaptation to culture is a narrow point of view to culture implications on software.

Culture affects the user's axis in Cameleon reference framework because culture implications are based on culture groups of users. Adaptation to users is wider than culture studies identify common behavior and practices of the user in his culture group. Anyway, it is also difficult to draw a line that separates adaptation to users from adaptation to culture. The difficulty is coming from the fact that in software we usually abstract common behavior of candidate users. This abstraction may be looked at as a creation of a culture group itself.

Culture can be defined at national and even regional levels according to Hofstede. We may expect that adaptation to environment to be part of adaptation to culture. Some regions are hot and mist (India). Others are dusty (desert). Such environment conditions may affect context of use and hardware equipment selection. Michael et al in (Michael, 2011) present the case of the Warana Wired Village project in a farm in India. They see the project as an example of technological breakdown. The project was aimed to enable farmers to communicate collaboratively. PCs used in the project were affected by environment conditions and cables were cut by rats. The Warana Unwired project, that proposes mobile phones instead of PCs, replaced that project.

Platform axis maybe considered as influenced by culture. Gitau's study demonstrates that many vast number of Internet users in South Africa are accessing the Web either exclusively or primarily through mobile devices (Gitau and Donner, 2009). We may say that selection of a platform is a culture preference.

Implications of culture on context of use as defined by Cameleon Reference Framework can be visualized in figure 6.

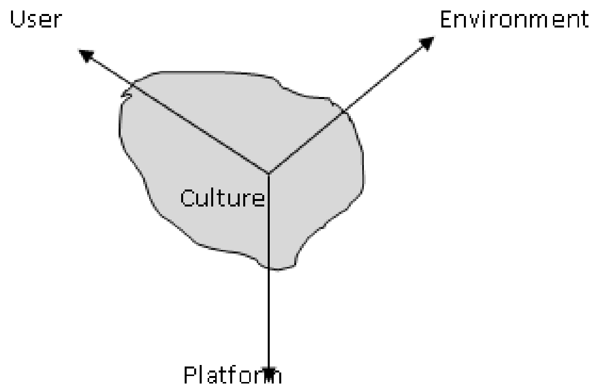


Figure 6. Culture scope according to Cameleon Reference Framework

Research methods could also be affected by culture like surveys and interviews. An ideal sampling techniques is to call participants randomly. Medhi et al (Medhi, 2007) explain that people in India may be less amenable to cold calls from strangers. Testing also is affected by culture backgrounds. Smith explains issue on global usability testing (Smith, 2011). Indians don't feel easy to say no or criticize the work of others. Chavan describes some methods to mitigate this discomfort felt by Indians in (Chavan, 2005).

Organizations form a level in Hofstede's model of culture. According to Hofstede, organizations represent a small society that develop its own culture values and practices. ERP customization to meet different organizations needs and internal methods falls in the category of implications of culture (at the organization level) on software development. Companies may follow the same general process but differ in some details and tasks. This difference may result from different regulations and legislations in the country (i.e. tax), from adapting different internal procedures the suit different situations (processing clients' orders differ according to type of business and technology infrastructure in the country). Some differences may be related to culture findings presented before like long hierarchies in organizations in high power distance countries vs short and flexible hierarchies in organizations with low power distance values.

Limiting culture implications to adaptation only (as modifications to an existing) is a restrictive point of view. With lack of information and universal culture model and ontology, we address culture from software development point of view as the need to enable users to implement creative culture solutions and enhance their ability to implement and manage changes.

Back to our question at the beginning, unfortunately, it is hard to identify if a change is related to culture or not. To be pragmatic again, we consider that any context of use is subject to a cultural situation, thus changes imposed on UI in this context can be related to a different culture group. This broad assumption may lead to situation where the culture group is arguable. For instance, if an application needs to support two types of mobile phones: touch screens and keypad. This change is considered as a change on the framework in Cameleon. For us, changes related to each type should be mapped to a culture group. Thus, we need to define the two culture groups: users with touch screen mobile phone and users with keypad mobile phones. We argue that People who chose mobiles with touch screen share the same preferences and practices for mobiles.

2.6 Concerns and software architecture culture requirements

Concerns on culture knowledge:

The literature review shows that culture knowledge are (1) scattered: mainly this is due to the wide range of implications. (2): domain-dependent: validity of guidelines in other domains may be questioned.

These concerns are magnified when we look at critics on culture studies themselves. Henrich provides some statistics based on a large literature review on culture studies and behavioral studies (Henrich, 2010); the bases for studies on culture implications on HCI, He states that "..analysis on the top journals sin six sub-disciplines of psychology from 2003 to 2007 revealed that 68% of subjects came from United States, 96% of subjects where from Western Industrialized countries… The make-up of these samples appears to largely reflect the country of residence of the authors, as 73% of first authors were at American universities, and 99% were at universities in Western countries. This means that 96% of psychological samples come from countries with only 12% of the world's population."

Unfortunately, we can do nothing in this regard. We consider these limitations on culture knowledge as a justification for our approach to technically support UI developers, leaving validity of culture implications to their own judgment.

Concerns on implementing culture knowledge:

If a UI developer has trusted culture knowledge; like explicit culture knowledge, what challenges may appear?

In order to understand challenges and concerns, and thus requirements, towards a CA-UI architecture, we present an example that explains how software UI evolves with the need to support new culture groups and consequent changes.

For simplicity, we work on supporting the guideline on adapting user interfaces to user's culture: units and measures used in the program should adhere to the user's culture. This guideline is implemented in many e-commerce web sites and in operating systems (like windows) using different implementation styles. Although this guideline is clear as a requirement, implementation is not that evident. We need to consider the architecture of the target software and also adopt a specific style.

Assume a company that develops a B-B e-commerce where sellers and buyers communicate and perform trading transactions. The first implementation of the B-B software product is designed to be used on a national level. The product employs only one currency that can be configured in the settings. Implementing the culture guideline above can take several styles, driven by innovation:

1. National level: sellers and buyers use the same national currency and don't need another. The guideline is already implemented in the first version of the product and no further changes are needed.
2. International level: If the company wants to sell the product to an international broker, international currencies should be employed (Us Dollar or Euro). Using one of these currencies is a practice of international sellers and buyers. The company then needs to update the product to suite this identified culture group 'international sellers and buyers'. One solution is to create a user's profile with the preferred currency. Buyers will see converted prices according to their preference. Sellers can enter prices according to their preference also. The configuration value is no more needed.
3. International level with both currencies: Some sellers accept to sell products in both currencies (USD and EUR). Some buyers prefer to buy products with offered currency to avoid conversion fees. Both buyers and sellers may also have a preference for one of these currencies. One style is to show prices in both currencies. This style might be refined into one of two styles: display a combo-box with both currencies with preferred currency as the default and a mark for the original price. The other refined style is to show both currencies at the same time with needed marks on prices. For sellers, a combo-box or a radio group may be used to allow filling prices with desired currency. Seller's currency preference can be the default choice.
4. International mobile users with both currencies: An international B-B broker needs to support end users with mobile phones. If the company implemented the refined style with showing price in both currencies, it may revert to the one with combo-boxes to better fit on mobile's small screens.

Our concerns on implementing the guideline in this example are:

Concern 1: The company would like to have the ability to sell the product to the broadest range of clients. Each style above fits a group of users. Supporting all styles in a single software package may become complicated if we consider other requirements for these groups. Creating a version for each group would increase maintenance and updates efforts.

Concern 2: The same culture knowledge (the culture guideline) can have several implementation styles depending on preferences of the culture group of users. Some styles may need more refinement to be implemented. A hierarchy can be noticed in style 3 “International level with both currencies”. The refined styles: “use combo-box” and “show both” are to sub-styles of the parent style “show both currencies”. As these styles reflect change requirements on the UI, a classification of these changes may help in implementation if that classification could reflect the place to implement the change. The hierarchy of implementation styles for the guideline is represented in figure 7. The leaves in the hierarchy are related to culture groups. Number of leaves represents the number of culture groups in the UI application.

Concern 3: A sub group of a culture group may not follow the same preferences. The culture group for the “International level with both currencies” style is a sub-group of “International level”, but their preferences differ. Thus, we can’t always rely on styles that suite a group of culture for its sub groups.

Concern 4: The first style shows that some culture guidelines might be implemented in the UI implicitly. Developers need to start with assumptions about the context of use. Some of these assumptions are about culture. Sometimes, it is difficult to separate application requirements from adaptation requirements. One may argue in the example above that use of international currency is a requirement in the application and should not be treated as part of adaptation to culture. We believe UI applications need to have a set of culture assumptions or else it is hard to image how to be created. Reality represented in HCI applications should reflect part of human life and practices. For instance, it is hard to imagine an application UI without using a human language for the output. Thus, culture is inevitably there. The point is: developers don’t know that culture is there until they know about a different culture. In this case, developers may isolate the part of code that is related culture difference and modify it with code that fits the other culture. As a result, developers are turning implicit culture guideline style into an explicit one.

Concern 5: suggested implementation styles are highly dependent on the domain, the specific application, and even sometimes on the UI widget itself. If the UI widget displays another element that shows a price (fees for taxes or shipping) the style may change (the original currency for shipping may differ from that for the product).

Based on concerns above, requirement to support developing adaptable to culture UI are:

Requirement 1: Because we develop software UI based on implicit assumptions about culture, an iterative-refined approach to develop software UI. This requirement addresses partially concern 4.

Requirement 2: an extensible architecture that allows supporting newly identified culture groups as extensions to the system. This extensible architecture addresses concern 1 because the company (in the example) has the ability to support all culture groups (and thus clients) by processing the required extensions. (Concern 1)

Requirement 3: a hierarchal classification of guideline implementation styles based on the type of change. This classification should help in identifying the place of change in the code. (Concern 2)

Requirement 4: a flexible UI architecture that enhances ability to modify the code (Concern 3 and 4). Enhancing ability can result from requirement 3 if classification can relate implementation style with fragment of code needed. The UI architecture should be flexible to support such relation and changes to fragments. In this sense, a flexible architecture helps also in extracting implicit culture guideline implementation style and presenting it explicitly.

Requirement 5: Enhancing traceability in architecture. To address concern 5, we need to trace relation between concrete UI elements and higher conceptual design elements (semantics and maybe goals).

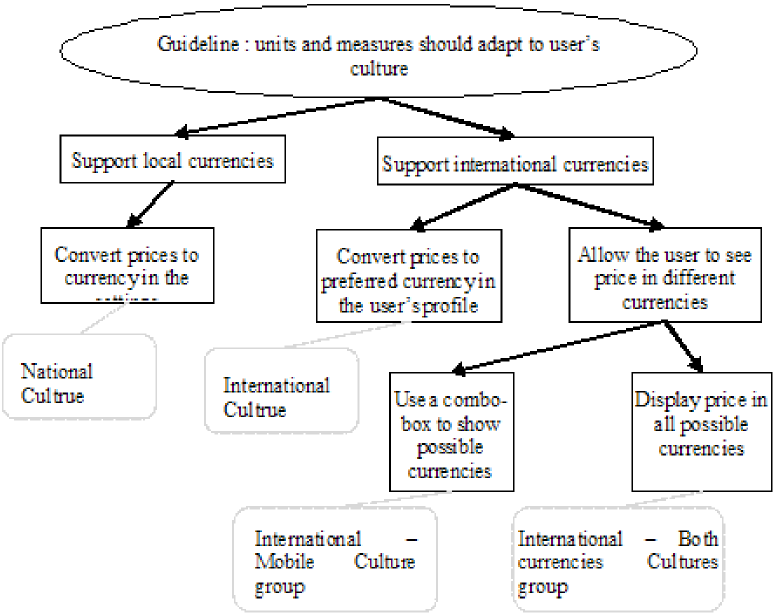

Guideline : units and measures should adapt to user's culture
Support local currencies
Support international currencies
Convert prices to currency in the
Convert prices to preferred currency in the user's profile
Allow the user to see price in different currencies
National Cultrue
International Cultrue
Use a combo-box to show possible currencies
Display price in all possible currencies
International – Mobile Culture group
International – Both currencies Cultures group

Figure 7. A hierarchy of implementation styles for a culture guideline

An architecture that satisfies the above requirements may be considered as CA-UI architecture. In the rest of this article, we will present an architecture that addresses some of these requirements. Requirements 4 and 5 are tackled in the discussion in section 4, while requirement 3 is partially considered in our proposed architecture.

3. Towards Culture-Adaptable UI architecture

3.1. Analyzing culture changes

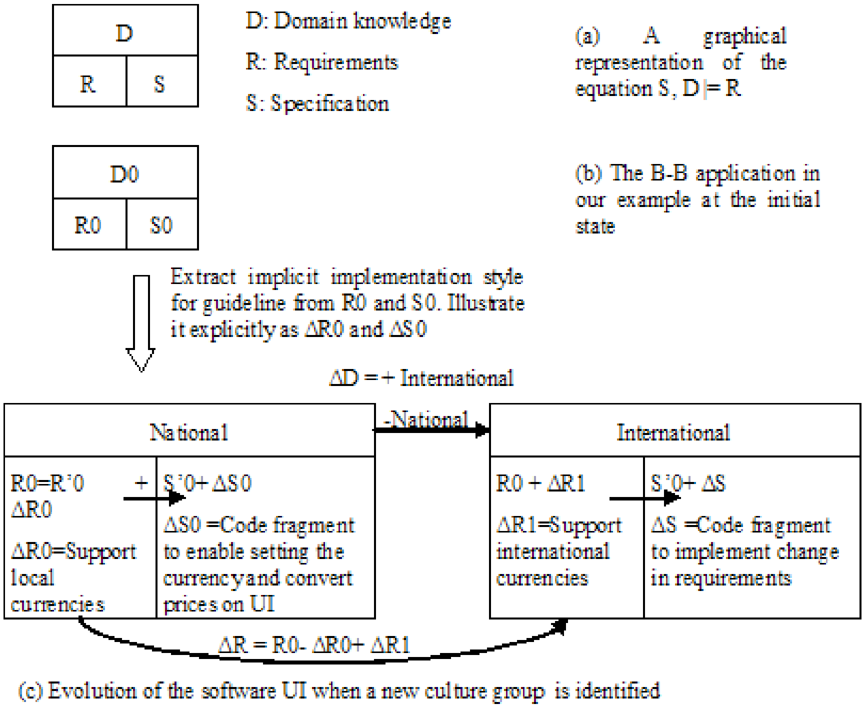
Our approach to create CA-UI architecture is based on supporting changes imposed by culture. To build the global picture of architecture, we analyze these changes using the “Reference Framework for Change and Adaptation” proposed by Filieri in (Filieri, 2012). This reference framework is created to allow precise talking about changes and adaptation. It is proposed in the domain of software engineering, so some terms may differ from terms used in Human Computer Interaction (HCI). The best way to describe the reference framework is to borrow the authors own words which we present below in italics.

The software developed by engineers provides an abstract machine whose goal is to fulfill certain requirements in the world in which it is embedded. M. Jackson and P. Zave, in their seminal work on requirements, clearly distinguish between the world and the machine. The machine is the system to be developed; the world (the environment) is the portion of the real-world affected by the machine. The ultimate purpose of the machine -the requirements-are always to be found in the world. Thus they are expressed in terms of the phenomena occurring in the world, as opposed to phenomena occurring inside the machine. Some of the world phenomena are shared with the machine: they are either controlled by the world and observed by the machine, or controlled by the machine and observed by the world. A specification (for the machine) is a set of prescriptive statements on the relation on shared phenomena that must be enforced by the system to be developed. To achieve requirements satisfaction, the machine relies on assumptions on the environment, captured as domain knowledge. "The primary role of domain knowledge is to bridge the gap between requirements and specifications."

Domain knowledge captures the set of relevant assumptions that need be made about the environment in which the machine is expected to work in order to be able to prove that (through the machine) we achieve the desired requirements. Let R and S be prescriptive statements in some formal notation that describe the requirements and the specification, respectively, and let D be the descriptive formal statements that specify the domain assumptions. If S and D are all satisfied and consistent, then it should be possible to prove that R also holds:

$S, D \models R$

That is, S ensures satisfaction of the requirements R in the context of the domain properties D.


D
R
S
D: Domain knowledge
R: Requirements
S: Specification
D0
R0
S0
Extract implicit implementation style for guideline from R0 and S0. Illustrate it explicitly as ΔR0 and ΔS0
ΔD = + International
-National
National
R0=R'0 + ΔR0
ΔR0=Support local currencies
S'0+ ΔS0
ΔS0 =Code fragment to enable setting the currency and convert prices on UI
International
R0 + ΔR1
ΔR1=Support international currencies
S'0+ ΔS
ΔS =Code fragment to implement change in requirements
ΔR = R0- ΔR0+ ΔR1

(a) A graphical representation of the equation S, D |= R

(b) The B-B application in our example at the initial state

(c) Evolution of the software UI when a new culture group is identified

Figure 8. B-B example represented after the reference framework for change and adaptation

In order to use this framework, we need first to map terms to HCI domain terms. The domain knowledge in our case is the culture knowledge represented as culture guidelines. The domain assumptions about the environment are culture groups. Our definition of culture groups in section 2.4 covers Cameleon's context of use. Requirements represent guideline implementation styles (changes on UI to support a culture group). Specification represents code fragments to implement the implementation style.

This reference framework can help in understanding changes in our example as follows: The original version of the e-commerce solution of the company contains: The application (specification) S0, the original requirements R0 in the original context of use D0. The new domain knowledge is introduced in the guideline. This leads to change in requirements R

We interpret supporting culture implication on UI after this framework as: when the UI developer acquires new culture knowledge, she/he may define a new culture group ΔD (update the domain assumptions) and adds needed requirements ΔR to update the UI. Consequently, she/he adds needed modifications ΔS to the UI on the program. Our example can be illustrated as in Figure 8.

3.2. The proposed architecture

In figure 7, we presented a hierarchy of guideline implementation styles. We mapped each culture group to a leaf. We re-draw figure 8.c to introduce this hierarchy and thus move culture groups from the representation for simplicity. The result in figure 9 is an extensible architecture that satisfies requirement 2. The architecture shows how the UI evolves when a new culture group is identified. For each group, an extension is added. This extension has two parts: the requirements, coming from the guideline implementation style selected, and the specification, the code fragment that implements the change in requirement. The evolution reflected in this architecture allows employing this architecture in an iterative method (requirement 1). We should note that from the beginning of development, every requirement should be treated as if it belongs to a culture group. Thus every requirement is related to a code fragment. When a requirement is found to be different for a culture group, it can be easily replaced by an appropriate one. Requirements here are detailed requirements of the change, and not only high-level ones.

Classification of implementation styles is based on refining an abstract change into a more concrete one. It is incorporated in the architecture. Thus, it partially satisfies requirement 3. The missing property in the classification is the ability to be used to decompose the specification. A classification that satisfies requirement 3 should be usable on both sides of the architecture: requirements and specification.

Our proposed architecture satisfies some requirements. It provides the bases and the global view towards the architecture we are looking for; an architecture that satisfies all of the 5 requirements.

4. Discussion

We believe classification is the key issue to satisfy our CA-UI architecture requirements. This classification can be employed in classifying changes and at the same time in decomposing the specification part in our architecture into several levels.

Nielsen's model introduced in (Nielsen, 1986) employs and linguistic classification for decomposing its model into 7 levels depicted in figure 10. These levels are:

Level 7 (Goal): expresses a user's mental goal, such as "search for a particular customer having a water meter in a specific region".

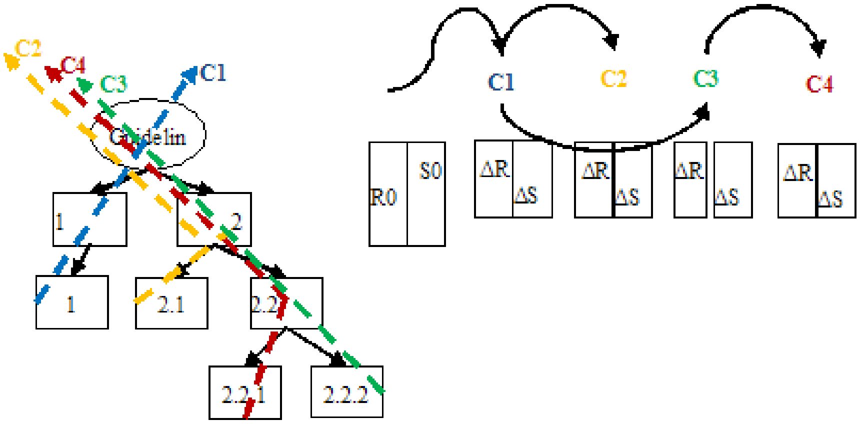
Level 6 (Pragmatic/task): translates this mental goal into a task to be carried out in the system according to the system concepts, such as "search for a subscriber having at least one water meter in zone x".

Level 5 (Semantic): translates the real-world objects into system objects and functions, such as "search for a subscriber with a code region filled in".

Level 4 (Syntactic): structures the semantic into an ordered sequence of operations in time and space, such as "select a zone code from the list and launch a query".

Level 3 (Lexical): decomposes each operation into the smallest possible pieces of information, such as "a zone code".

Level 2 (Alphabetic): specifies the unit of information (e.g., a lexeme, a metric) for each information item, such as "an integer for representing the zone code".



Hierarchy of implementation styles for the guideline

A possible evolution of software UI

1: Support local currencies
1.1: Convert prices to currency in the settings
2. Support international currencies
2.1 Convert prices to preferred currency in the user's profile
2.2 Allow the user to see price in different currencies
2.2.1 Use a combo-box to show possible currencies
2.2.2 Display price in all possible currencies

C1 : National Cultrue group
C2 International Cultrue group
C3: International –Both currencies Cultrue group
C4 : International –Mobile Cultrue group

Figure 9. Architectural overview of the UI evolution.

Level 1 (Physical): specifies the physically-coded information in terms of light, sound, color, etc., such as "colors emitting from the screen, sound waves coming from the speakers".

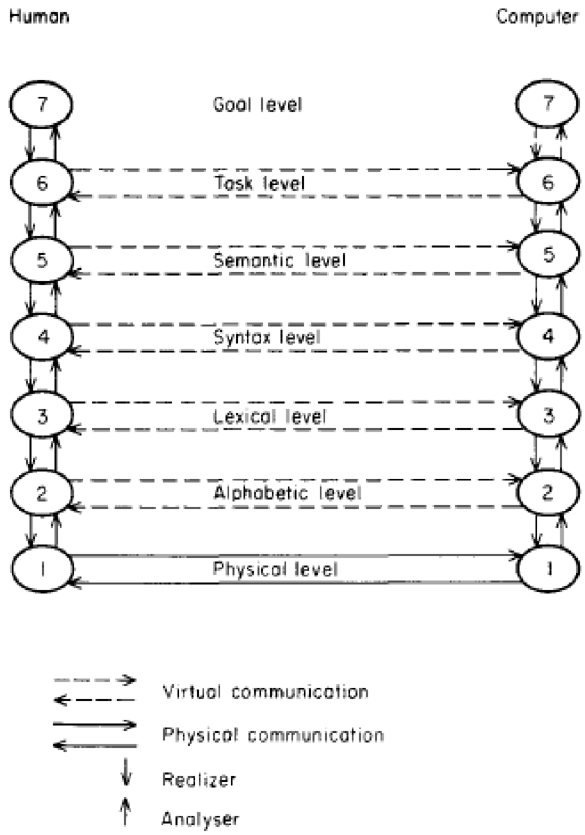
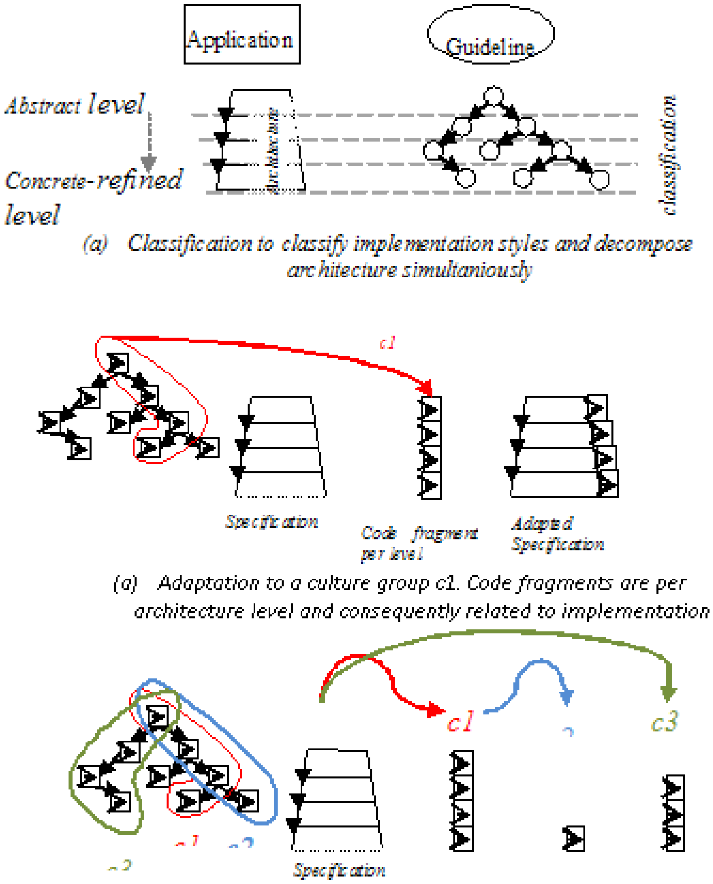


Figure 10. Overview of Nielsen's Virtual Protocol model.

Each level of Nielsen's model is concerned with different aspects than the others. At the same time, each level is a refinement of the upper level. Our idea is the following: assume we could define a model at each level of Nielsen's model. Then use the linguistic criteria to classify changes imposed by implementation styles on UI. For instance, the implementation styles 2.2.1 and 2.2.2 in figure 9 differ in the type of UI control to use: the first suggests using a combo-box while the other suggests using another control, like a table to display prices in different currencies. Changing the type of UI control fits at a specific level (say lexical). Evolution from culture group C3 to C4 would require changing the model on lexical level only without affecting at least upper levels. This would result in increasing flexibility and enabling at the same time mapping implementation styles to specification code fragments more precisely. Figure 11 represents how the architecture would look like if Nielsen's model is employed.



(a) Classification to classify implementation styles and decompose architecture simultaniously

(a) Adaptation to a culture group c1. Code fragments are per architecture level and consequently related to implementation

(c) Evolution of UI application. Evolution may happen per level

Figure 11. Targeted Architectural overview.

Nielsen's model requires more work out and adaptation to be employed in the architecture. Linguistic classification also needs more adaptation and explanation how to be used to changes on the UI. Anyway, the idea is promising and **worth f**urther research.

6. Conclusion and Future Work

In this article, we addressed culture from the point of supporting implementing changes imposed by culture on UI. This approach is mainly technical and aims to support UI developers and designers in achieving their culture-adapted UI.

We presented a fair literature review on culture studies that served to motivate and justify our approach and to explain concerns imposed by adaptation to culture on UI development. We stated a set of architecture requirements to address these concerns and proposed an architecture that partially satisfies the stated requirements.

Some requirements are hard to satisfy. We tackled them by suggesting the idea to use linguistic classification to classify changes to UI. The idea is promising to address flexibility requirement, because it decomposes implementation styles and specifications in our suggested architecture and thus allows evolution with less changes. It may also address traceability because is relating implementation styles and fragments of code to on a more refined level.

In the future, we will investigate appropriate methods to adapt Nielsen’s model and linguistic classification to be integrated in our proposed architecture and thus satisfy further requirements if not all.

Although the main purpose of this work is proposing a CA-UI architecture, but if we could adapt linguistic classification properly, or even find another classification, we may expect another promising contribution coming from the classification of implementation styles. Such a classification may serve as a way to accumulate, classify and even exchange culture knowledge and their implementation styles. Accumulation and classification happens in an iterative way along the development and the evolution of the UI. As the classification proposed is a kind of formalism for these guideline changes that can be extracted from the architecture easily, it is possible to exchange and share this knowledge with others in the same domain of applications. This exchange can be seen as an effort to incorporate guidelines in UI development; a process that is referred to as operationalization (Vanderdonckt, 1999).

References

Ackerman, S. K. (2002). Mapping User Interface Design to Culture Dimensions. International Workshop on Internationalization of Products and Systems. Austin TX, USA.

Adler, N. J. (2002). International dimensions of organizational behaviour. Cincinati, Ohio: Thomson Leaning.

Callahan, E. (2005). Cultural similarities and differences in the design of university websites. *Journal of Computer-Mediated Communication, 11(1) , article 12*.

Calvary, G., Coutaz, J., Thevenin, D., Limbourg, Q., Bouillon, L., Vanderdonckt, J., A Unifying Reference Framework for Multi-Target User Interfaces, *Interacting with Computers, Vol. 15, No. 3*, June 2003, pp. 289-308

Chavan, A., L.: Another culture, another method. In: Proceedings of the Human Copmputer Interaction International Conference (2005).

Dormann, C., & Chisalita, C. (September 8-11, 2002). Cultural values in web site design. Eleventh European Conference on Cognitive Ergonomics. Catania, Italy.

Edwards, J., Bagozzi, R. (2000) On the nature and direction of relationship between constructs and measures. *Psycological Methods 5(2)*, 155-174.

Evers, V., & Day, D. (July 14-18, 1997). The role of culture in interface acceptance. *Conf. on Human-Computer Interaction INTERACT'97 (pp. 260–267)*. Sydney: Chapman & Hall, London.

Filieri, A., Ghezzi, C., Tamburrelli, G: A formal approach to adaptive software: continuous assurance of non-functional requirements, Formal Aspects of Computing, 24, 2, March 2012.

Fogg, B. J., and D. Lizawa. 2008. Online persuasion in Facebook and Mixi: A cross-cultural comparison. Paper presented at the 3rd International Conference on Persuasive Technology, Oulu, Finland, June 4–6.

Gitau,S., Donner,J.: New paths: exploring mobile-only and mobile primary internet use in South Africa. Presented at W3C workshop on the Africa Perspective on the Role of Mobile Technologies in Fostering Social Development, Maputo (2009).

Hall, E. T. (1989). Beyond culture. New York: Doubleday.

Henrich, j., Heine, S., Norenzayan, A. (2010): The weirdest people in the world?. Behavioral and Brain Sciences, 33, 61-83.

Hofstede, G. (1997). Culture and organisation: Software of the Mind. New York: Mc-Graw Hill.

Kluckhohn, F. R., & Strodtbeck, F. L. (1961). Variations in value orientation. Peterson.

Marcus, A. (19-24 July 2009). The impact of Culure on design of Arabic Websites. Proc. HCI International Conference. Town and Country Resort and Convention Center, San Diego, CA: Springer Verlag.

Marcus, A., & Krishnamurthi, N. (2009). Cross-Cultural Analysis of Social Network Services in Japan, Korea, and the USA. (N. Aykin, Ed.) Internationalization, Design and

Global Development , 59-68.

Marcus, A., & West Gould, E. (July 2000). Cultural Dimensions and Global Web User-Interface Design. Interactions 7, (pp. 32–46).

Medhi, I., Sagar, A., Toyama, K.: Text-free user interfaces for illiterate and semi-literate users. Information Technologies and International Development. P.37-50. 2007.

Michael L. Best and Thomas N.Smyth: Global/Local Usability: Locally Contextualized Usability in the Global South. Global Usability, Human-Computer Interaction Series. Springer-Verlag, London 2011. P.9-22. ISBN: 978-0-85729-303-9.

Minocha, S., T. French, and A. Smith. 2002. eFinance localisation: an informal analysis of specific eCulture attractors in selected Indian and Taiwanese sites. In Proceedings of the 4th International Workshop on Internationalization of Products and Systems, ed. D. Day.

Meyer, B. On formalism in specifications. IEEE Software, January 1985

Nawaz, A., & Clemmensen, T. (2007). among, Cultural differences in the structure of categories. The Seventh Danish HCI. IT University, Copenhagen, Denmark.

Nielsen, J. A. Virtual Protocol Model for Computer-Human Interaction. International Journal of Man-Machine Studies, March 1986, Vol.24, Issue.3 , pp. 301-312.

Nisbett, R. E., & Masuda, T. (2003, 9 16). Cultural and point of view. 100.

OMG. (2001). Model Driven Architecture (MDA).

Rau, P.-L. P., Plocher, T., & Chong, Y.-Y. (2011). Cross-Cultural Web Design. In Handbook of Human Factors in Web Design. Second Edition (Ch.33. pp.677-698). London: Tylor & Francics Group, CRC Press.

Rau, P.-L. P., and S.-F. M. Liang.(2003). A study of cultural effects on designing user interface for a Web-based service. International Journal of Services Technology and Management 4(4–6): pp. 480–493.

Rajkumar, S. (2003). University Web Sites: Design Differences and Reflections of Culture. Unpublished manuscript, School of Library and Information Science, Indiana University Bloomington .

Reinecke, K. B. (2010). Modeling a User's Culture. In E. Blanchard, & D. Allard (Eds.), Handbook of Research on Culturally-Aware Information Technology: Perspectives and Models , pp. pp. 242-264.

Robbins, S. S., & & Stylianou, A. C. (May 2010). A longitudinal study of cultural differences in global corporate web sites. Journal of International Business and Cultural Studies, Vol.3 .

Smith, A.: Issues in Adapting Usability Testing for Global Usability. Global Usability, Human-Computer Interaction Series. Springer-Verlag, London 2011. P.23-38. ISBN: 978-0-85729-303-9.

Prabhu, G., and D. Harel. 1999. GUI design preference validation for Japan and China—a case for KANSEI engineering. In Proceedings of the 8th International Conference on Human–Computer Interaction (HCI99) (Munich), 521–525.

Wind, J., Riege, K., Bogen M., 2007. Spinnstube®: A Seated Augmented Reality Display System, *Virtual Environments: Proceedings of IPT-EGVE – EG/ACM Symposium*, 17-23.

Vanderdonckt, J. (1999). Development Milestones towards a Tool for Working with Guidelines. *international Journal of Man-Machine Studies , 24 (3), 301-312*

Zhao, C. 2002. Effect of information structure on performance of information acquiring: A study exploring different time behavior: monochronicity/polychronicity. PhD diss., Institute of Psychology, Chinese Academy of Sciences, May