# Cultural Issues and Opportunities in Computing Education

Ben Keller, Manuel A. Pérez-Quiñones[1], Ravikiran Vatrapu[2]
Eastern Michigan University
Computer Science
511 Pray-Harrold
Ypsilanti, Mi 48197 bkeller@emich.edu

***Abstract*** **- In this paper, we present an argument about why cultural issues need to be included in Computing curricula. We briefly discuss some of the literature in cultural issues and identify how it is related to different areas of computing. Then we identify opportunities in CC2001 that allow the inclusion of cultural discussions in our courses.**

*Index Terms* - Globalization, Internationalization, Curriculum, Cultural Issues in Computing.

## INTRODUCTION

Culture refers to particular patterns of behavior that identify a group of people. Culture has an impact on everything we do; how we think, how we interact and communicate with others, how we process information, even how we design artifacts and systems. It is no surprise then that cultural issues should be a common theme in all of higher education.

In computing, cultural issues used to be relevant only when building systems that were to be deployed to other countries. However, with today's global communication networks, global economy, and globally accessible internet content and applications, cultural issues take on a major role in computing.

In this paper we present the argument that the computing curricula [8] needs to be more inclusive with issues of culture. We argue that cultural issues are pervasive in computing, and that they affect data representation, database design, communication protocols, software engineering methods, and user interface designs among other areas of the curriculum. It is our belief that we cannot afford to treat cultural issues as a unit to be covered in a (mostly elective) Human-Computer Interaction course.

Our goal goes beyond influencing the next round of curriculum committees. We present opportunities to introduce cultural issues across the curriculum. We hope that this will inform and motivate instructors to include material that addresses cultural issues in their courses.

## CULTURAL ISSUES IN COMPUTER SCIENCE

The need for coverage of cultural issues in computer science education is ultimately driven by the development of the global economy in which software systems are used around the world. Current curricula generally relegate these subjects to HCI courses, or to specialized courses only taken by a relatively small group of students, while cultural issues pervade the field. Beyond the obvious issue of language, culture impacts how even simple information such as dates and numbers is presented, what colors and symbols are appropriate on a website, and how well software development and evaluation procedures work.

Successful software systems now must be written so that adapting them to a particular culture can be done easily. This is the idea of globalization/localization in which software is built in such a way that localizing it to a particular language and other cultural preferences can be done easily, possibly at runtime by reading a user's profile. Building software in this way requires an awareness that cultural differences exist, some idea of what those differences are, and an understanding of design strategies for building software that is adaptable. The construction of web-based systems has a similar requirement, since these systems may present themselves in different ways depending on the localization and language preferences of the user.

Current systems yield examples of cautionary tales in dealing with culture that only serve to emphasize the need for coverage of cultural issues in computer science education. For instance, databases that require names to consist of a first, middle and last name (imagine being asked to legally change your name because it doesn't match the format expected by your employer's database [24]); online conference submission forms that cannot accept international characters; mail transfer protocols unable to transmit messages with international characters; and news websites that deliver news from Spain when the user selects Spanish as their language. These problems cover the full range of software systems, and make it clear that, at the very least, our students need to be made aware that cultural differences do exist, and need to be taken into account in their systems.

[1] Manuel A. Pérez-Quiñones, Department of Computer Science, Virginia Tech, Blacksbug, VA perez@cs.vt.edu
[2] Ravikiran Vatrapu, Communication and Information Sciences, University of Hawaii at Manoa, Honolulu, HI, ravikira@hawaii.edu

## BACKGROUND

*Definition of Culture*

Culture has been defined in many different ways by different researchers. A compiled list of over 200 different definitions of culture can be found in Kroeber and Kluckhohn's book [10]. In an editorial to a Special Issue of *Interacting with Computers* titled 'Global human-computer systems: cultural determinants of usability', Smith and Yetin state that researchers often refer to culture as "*patterns of values, attitudes, and behaviors, which are shared by two or more people*" [21]. A commonly used definition of culture is that of Geert Hofstede: "*Culture is the collective programming of the mind which distinguishes the members of one group or category of people from another*" [6]. Hofstede's cultural model is well suited for empirical research as a score for each individual member of the culture can be computed unambiguously.

*Hofstede's Cultural Model*

A particular model of culture that is easy to understand and that has origins in a study done with IBM employees is the one created by Geer Hofstede [6]. Hofstede's cultural model consists of five dimensions. Each dimension groups together phenomena in a society that were empirically found to occur in combination. Hofstede's seminal work on cultures in organizations formulated a framework of five dimensions of culture identified across nations. The five dimensions are: power-distance, collectivism-individualism, uncertainty avoidance, femininity-masculinity, and long-short term orientation. Hofstede formulated the dimensions from statistical analysis of the answers to questions provided by IBM employees in over 50 countries.

In Hofstede's work, all dimensions are represented by a single score. The cultures/countries are then ranked from large to small based on the score. The magnitude in a particular dimension scale is usually not that important, the importance is the ranking within the scale. As an example, India is a large power distance culture, and is ranked 10th/11th (tied with West Africa). The US is considered a small power distance country and is ranked 38th.

*Usability of Systems and Culture*

One of the areas of computing where the impact of culture is actively being studied is in Human-Computer Interaction. For example, the research done by Cliff Nass [13][14][15][16] in social aspects of HCI has shown that even computer-literate users tend to use social rules and display social behavior in their interactions with computers. And, social behavior is strongly grounded in culture as every person carries within himself or herself patterns of thinking, feeling and potential acting. Much of this is learned during early childhood. To learn new patterns of thinking, feeling and acting one has to unlearn the old patterns, which is more difficult than learning them for the first time [6].

Also international usability testing is inherently cross-cultural. Furthermore, it is conducted in a social setting. Thus, the thinking, feeling, perception and reactions of users during international usability testing are influenced by the participant's cultural and social background, and thus must be considered when evaluating the validity of usability engineering methods.

Nielsen [17] recommends traveling to the target country and conducting usability tests as the best choice in international usability testing.

*Culture and Usability Methods*

Even software development methods used are influenced by culture. Beu et al. [1] emphasize that explication and understanding in a foreign cultural context is only possible if there is intense cooperation between representatives of the different cultures. Beu et al. [1] report that there were problems when data from different cultural sectors of China had to be compared to draw conclusions about the design. In China despite the fact that participant profiles had been drawn up invitations had to be sent out to decision makers instead of end users. This was along the lines of the Chinese notion of hierarchy. The quality of the usability tests and the focus group discussions varied greatly depending on the discussion leader and the setting. This is explained mostly by the differences in the way people from different cultures work.

Yeo [25] described a study conducted to examine the efficacy of the global-software development lifecycle (global-SDLC), a Western software development approach employed to derive software for the global market. The results of the usability evaluation were found to be inconsistent. Yeo attributes inconsistencies to the large power distance and collectivist culture of Malaysia. The author says that it would appear that the inconsistencies stem from the participants' reluctance in providing critical negative comments, because of preservation of face and respect for hierarchy.

Marcus and Gould [12] applied Hofstede's cultural dimensions to web and user-interface design. The authors mention each of Hofstede's five cultural dimensions and the aspects of design that can be influenced by that particular dimension. They present screen shots of different web sites developed in different nations and point out the cultural influences on design. The findings amplify the cultural differences but are with out empirical evidence.

Honold examined the notion of culture and its relevance to Human-Computer Interaction and discussed the theories of culture in HCI [7]. Honold found cultural influences when a washing machine developed in Germany was used in India. Honold identifies several cultural factors that have to be taken into consideration in any investigation of the context in which the product is used.

Day and Evers [2] found cultural-based differences in interface acceptance. Results reported by Evers et al. [3][4] indicate that cultural aspects led to differences in user's expectations and understanding of the website of a virtual campus.

Sears et al. [20] examined the international differences and effect of high-end graphical enhancements on the perceived usability of the World Wide Web. They found

significant differences between the users belonging to the two different cultures of United States of America and Switzerland.

*Icons, Layout, Localization*

Fernandes [5] has identified various cultural issues of nationalism, language, social context, time, and currency, units of measure, cultural values, body positions, symbols and esthetics that need to be addressed during global interface design. Russo and Boor [19] present a checklist of cross-cultural items to be considered in interface design.

Khaslavsky [9] describes the impact of culture on usability and design, presents variables useful for incorporating culture into design and various design implications and mentions issues in localization of design. Development of software for international use is mostly done by internationalization and localization [11][22][23]. Symbols, heroes, rituals, values and practices are the most important manifestations of culture [6].

*Beyond Surface Issues*

However, the cultural issues identified by all these researchers consider only the symbols and rituals of different cultures. The impact of culture goes beyond icon use and user interface layout issues. We present three examples that are typical indicators of the breakdown that occurs when culture is not considered in the design of systems.

*Character sets.* Many cultures use extended character sets in the language that is characteristic of them. For example, Spanish has a reverse exclamation character (¡ or ¿) that appears at the beginning of all exclamation phrases. Spanish also uses tildes over the letter n (ñ) and accents over vowels (á, é, etc.). Failure to understand this, or assumptions that the removal of these "little characters" won't have a meaning impact in words can be disastrous. For example, words change meaning if the tilde over the n is removed, and the meaning could have very unintended effect. For example a word for pony tail in Spanish is *moño*, but if you remove the tilde you get *mono*, which means monkey. As a recent relevant example, the SIGCSE submission site for this year's conference does not accept accents and tilde in the name of authors.

Furthermore, some older communication protocols did not allow the use of extended character sets because they were strictly based on ASCII. This has been alleviated by the use of other standard representations that even if based on ASCII, they allow multi-character representations for international character sets.

*Culture and Language*. The relationship of culture and language is a very complex one. For example, it is often assumed that if you live in the US you speak English. But this is not always true; consider immigrants or workers that live in the US for a short period of time on some professional capacity (e.g. diplomats).

When software tries to match language and culture, the implications can be confusing. Consider Google's news page. At the time of this writing, if you select a country from the "Top Stories" menu, you will be shown news from that country. Furthermore, the language of news is automatically switched to the *accepted* language of the country. What if a person was planning a trip to Spain and wanted to read up on the current news from Spain. Selecting "Top Stories Spain" would automatically switch to Spanish, a language that the user might not be familiar with. Furthermore, the news section titled "International" changes meaning. For the US Top Stories, international means stories from outside of the US. For the Spain Top Stories, international means outside of Spain.

*Personal Names.* Different countries have different conventions when it comes to writing names. But one thing that has been very US centric is the exclusion of spaces in the names. There are plenty of cultures that use last names which use multiple words [18]. The simplistic design that spaces to separate names is inappropriate. Designers must use different ways to separate fields in data representations, spaces is not a valid way to separate them without excluding some of the world population.

*What is being done*. We solicited information on the inclusion of cultural issues in courses on the SIGCSE mailing list. What the responses indicate is that most of the effort is specific to specialized courses on Globalization and Internationalization, HCI courses, and web-development courses. As far as we know, no program has fully integrated cultural issues into the curriculum.

Some cultural issues are already present in the CC2001 Computer Science Curriculum in the HCI, Graphics and Visual Computing (GV), and Society and Professionalism (SP) bodies of knowledge. In particular, the influence of culture on graphical and GUI design is dealt with in HC1 Foundations of Human-Computer Interaction and GV3 Graphic Communication. And, the intercultural implications of computing on civil liberties and privacy are dealt with in SP7 (Privacy and Civil Liberties). International issues are also addressed in SP2 (Social Context of Computing) and SP6 (Intellectual Property). Note that the substantial coverage of culture impacts on developing systems is restricted to the HC body of knowledge, which is typically treated as an elective. Since, no other explicit mention of cultural issues is made, the current curriculum doesn't adequately address the concerns we have raised.

## OPPORTUNITIES IN CURRICULUM

Ultimately, our goals in addressing culture in the curriculum are to allow students to

- Build systems that are adaptive to the cultural preferences of a user; and
- Be able to work effectively in an environment where differences in culture exist.

Therefore, we consider opportunities that fit into three categories:

1. Using globalization/localization technology to build software that is adaptive to cultural preferences of the user.

2. Building technology that supports globalization and localization
3. Understanding the influence of culture on our professional interactions

We outline the basic opportunities in these three categories and associate them with the bodies of knowledge in the CC2001 Computer Science Curriculum [8]. Note that in some of these cases we present possible programming assignments or homeworks. More importantly, in our opinion, is the dialogue that will develop as students encounter some of these cultural situations. The most important aspect of inclusion of cultural issues is raising awareness in the student about the cultural differences that exist.

*Designing and Building Systems*

Much of the core computer science curriculum deals with the bodies of knowledge related to designing and building systems. Here we concentrate on opportunities that relate to the use of technology that supports different languages, cultural differences in information presentation, and locale-based and user preferences.

### INFORMATION REPRESENTATION

As mentioned earlier, different cultures present common information such as dates, times and numbers in different ways, and clearly different countries use different currency. Since modern programming languages, such as Java, support output of these values in different formats, it is fairly straightforward to print, say, a date in a culture specific format. A related opportunity is to have students do output of the results of their program is different formats. For instance, computing change in Euros, or Chinese Yuan using simple I/O (PF1 Fundamental programming constructs).

Another place where cultures differ in the representation of information is the names of people. In some cultures, people may only have one name, and in others they may have four. A typical assumption in the United States is that everyone has a first, last and middle name, which the authors have used in several classes as an example of invalid assumptions in software design. Opportunities are to design and build software (SE1) or databases (IM6) where different name conventions are allowed.

### SUPPORT FOR LANGUAGES

The support for output in alternative character sets is also important to building software that can be localized, and most introductory textbooks mention Unicode support. However, they do not go into detail about issues related to other languages. Specific issues that present opportunities are the fact that different languages may not use white space for word breaks (e.g., Chinese), or use a left-to-right script system (e.g., Arabic). These issues relate to strings and string processing (PF3 Fundamental data structures), and pattern matching and text algorithms (AL2 Algorithmic strategies). For string processing, an exercise would be to build a search and replace mechanism that works regardless of language; and for pattern matching, students might implement the Boyer/Moore algorithm(s) and experiment with performance on Asian versus European languages. These issues also present an opportunity to illustrate the dangers of making invalid assumptions, something that is important without consideration of culture.

### EMPLOYING LOCALES

A locale encodes the preferences of a particular location and determines the languages and presentation of information. In Java, a locale also includes an encoding of language/culture specific alphabetic comparisons. Use of language or library support for locales, allows the construction of comparison-based data structures (PF3 Fundamental data structures), and sorting and searching algorithms (AL3 Fundamental computing algorithms) that are independent of these assumptions. The use of locales could be introduced in an intermediate programming course, and then expected in all subsequent programming courses.

### ADAPTING TO USER PREFERENCES

Some programming languages and operating system specific APIs allow the specification of user preferences via property files. Such mechanisms are critical to building systems that are adaptive to user preference, and therefore cultural differences; however, they are not explicitly dealt with in the curriculum. Possible places where they might be dealt with are SE1(Software Design) or SE2 (Using APIs). Again, the use of property files could be introduced in an intermediate course as part of the way programs should be written.

### DESIGNING PROGRAMS

Many of the opportunities above relate to basic support for globalization/localization. In the context of GUI, and more advanced system development there are other issues as well, such as managing text messages in different languages, and GUI layout and color scheme. Note that while the Human Computer (HC) body of knowledge deals extensively with consideration of cultural difference, it does not specifically address these issues in terms of design. Opportunities of addressing the construction of localizable user interfaces include HC2 (Building a simple GUI), HC4 (Human-centered software development), HC5 (Graphical user-interface design), HC6 (Graphical user-interface programming). Since globalization/localization affects other issues of design other opportunities can be found in SE1 (Software design), SE2 (Using APIs) and SE3 (Software tools and environments) which would involve using tools for building localized software. Clearly cultural issues are also prevalent in construction of web-based applications (NC5) as well. A particular opportunity relates to the fact that some web servers (e.g., Apache) have the ability to support multi-language sites (NC4).

### INFLUENCE OF LANGUAGE ON COMPUTATION

In addition to the opportunities already mentioned, considering computing over different languages poses different opportunities that relate mostly to different characteristics of language. In particular, those characteristics would affect pattern matching algorithms (AL2), compression (NC7), distribution of keys in hashing (PF3), and the choice of algorithm (AL3). Other areas where language has an influence on computation are natural language processing (IS7), cross-language information retrieval (IM11) and digital libraries (IM14).

*Supporting Technology*

In addition to building culturally adaptive systems, the topic of how technology that supports these systems is constructed can illustrate important issues in design of APIs, programming languages, operating systems, and database systems. Specific opportunities relate to the design of GUI APIs with mechanisms for localization; the support of character sets in programming language design (PL11); the influence of globalization/localization on operating system and API design (OS1,OS2). Note that the HCI body of knowledge currently has no coverage of the design of APIs for graphical user interfaces. Otherwise, in most cases, these opportunities are instances of topics already covered by the curriculum, and so are opportunities to raise awareness of the influence of culture.

*Influence of Culture*

Other opportunities exist in terms of how culture affects how users interact with and perceive software, and how cultural differences may affect professional interactions.

## INFLUENCE ON USERS

As we have noted, the curriculum already deals with the influence of culture on both graphical (GV3) and GUI (HC1) design. However, the impact of cultural perceptions of users on system evaluation is not dealt with explicitly, but could be integrated into HC3 (Human-centered software evaluation) and SP3 (Methods and tools of analysis). Related is the opportunity to identify invalid cultural assumptions encoded into systems (such as a news site assuming a user reading news in Spanish wants news from Spain), which relates to both SP2 (Social context of computing) and SP3 (Methods of analysis).

The curriculum also deals with international and cultural issues in privacy and civil liberties (SP7), where there is an opportunity to explore the influence of cultural perception of privacy on system design. An example, is the differing willingness of users to provide information to a recommender system. A related opportunity is the identification of barriers to cross-cultural communication in communication and collaboration systems, which relates to HC8 (HCI aspects of collaboration and communication) and SP2.

## PROFESSIONAL INTERACTIONS

It is clear that intercultural differences increasingly impact professional interactions as the globalization of the software industry grows. But this is not an issue dealt with directly by the bodies of knowledge, though the narrative of the curriculum does mention using interactions within team projects to model these scenarios. Certainly, there is an opportunity to deal with these issues in SE8 Software project management. Specific cultural differences that could cause problems in a professional setting are differences in perception of intellectual property rights (SP6), and privacy (SP7).

As noted earlier, another place where cultural differences impact professional interactions is in user interface evaluation. Coverage of strategies for cross-cultural evaluation are appropriate to HC8 Human-centered software evaluation.

## CONCLUSIONS

We argue that the infusion of cultural issues in computing education is important to the success of our students. In part, the goal is to make students aware of cultural differences and how they impact software development and use, and the interactions they are likely to have with others as professionals. Another goal is to teach students how to build software that can be adapted to the cultural preferences of users. The opportunities that we have outlined are of two types. The first is how cultural issues can be used to emphasize subjects that are already covered by the curriculum. Examples include identifying invalid design assumptions, and the use of APIs. The second are relatively new ideas, such as the cultural influence on user interface evaluation.

## REFERENCES


[1] Beu, A., Honold, P and Yuan, X. How to Build Up an Infrastructure for Intercultural Usability Engineering. *The International Journal of Human-Computer Interaction, Volume 12, Numbers 3&4*, pp.347-358, 2000.

[2] Day, D., and Evers, V. The Role of Culture in Interface Acceptance. *Human Computer Interaction*, INTERACT' 97.

[3] Evers, V. Cross-cultural Understanding of Metaphors in Interface Design. In *Ess, C. & Sudweeks, F., Attitudes toward Technology and Communication (proceedings)*, London, Aug. 1-3, 1998.

[4] Evers, V., Kukulska-Hulme, A and Jones, A. Cross-Cultural Understanding of Interface Design: A Cross-Cultural Analysis of Icon Recognition In *E. del Galdo and G. Prahbu (Eds.), Proceedings of the International Workshop on Internationalization of Products and Systems*. Rochester.

[5] Fernandes T. Global Interface Design. *Academic Press*, 1995

[6] Hofstede, G. Cultures and Organizations: Software of the Mind, Intercultural Cooperation and its Importance for Survival. *McGraw-Hill*, New York, NY, 1997.

[7] Honold, P. Culture and Context: An Empirical Study for the Development of a Framework for the Elicitation of Cultural Influence in Product Usage. *International Journal of Human-Computer Interaction, 12 (3 & 4)*, pp.327 – 345, 2000.

[8] The Joint Task Force on Computing Curricula (2001) Computing Curricula 2001, *Journal on Educational Resources in Computing*, v1, n3es. ACM Press, New York, NY, USA, pp. 1531-4278.

[9] Khaslavsky, J. Integrating Culture into Interface Design. *ACM CHI 98*, pp.365-366, 1998.

[10] Kroeber, A.L. and Kluckhohn, C. Culture: A Critical Review of Concepts and Definitions. *Harvard University Peabody Museum of American Archeology and Ethnology*, Volume 47, 1952.

[11] Luong, T., Lok, J., Lok, S., and Driscoll, K. Internationalization: Developing Software for Global Markets. *Wiley*, New York, 1995.

[12] Marcus A. and Gould, E.M. Cultural Dimensions and Global Web User-Interface Design. *Interactions* pp.32 – 46, July-August, 2000.

[13] Nass, C., Steuer, J. and Tauber, E. R. Computers are Social Actors. *Conference Proceedings on Human Factors in Computing Systems: "Celebrating Interdependence"*, pp. 72-78, 1994.

[14] Nass, C., Moon, Y., Fogg, B. J., Reeves, B and Dryer, C. Can Computer Personalities be Human Personalities? *Conference Companion on Human Factors in Computing Systems*, pp.228 – 229, 1995.

[15] Nass, C. and Moon, Y. Adaptive Agents and Personality Change: Complementarity versus Similarity as Forms of Adaptation. *Proceedings of the CHI '96 Conference Companion on Human Factors in Computing Systems: Common Ground*, pp.287–288, 1996.

[16] Nass, C., Kim, E-Y. and Lee, E -J. When My Face is the Interface: An Experimental Comparison of Interacting with One's Own Face or Someone Else's Face. *Conference Proceedings on Human Factors in Computing Systems*, pp.148-154, 1998.

[17] Nielsen, J. International Usability Engineering. *In Del Galdo, E. and Nielsen, J. (Eds.). International User Interfaces, John Wiley & Sons*, 1996.

[18] Pérez-Quiñones, M.A. (2002) Hispanic last names: Why two of them? Conductor, Publication of the Office of Multicultural Affairs, Virginia Tech, October 4, 2002. Available at http://www.multicultural.vt.edu/conductor-arch/con2002-1004.pdf (Last Accessed on 09-09-2005)

[19] Russo, P. and Boor, S. How Fluent is your Interface? Designing for International Users. *Proceedings of INTERCHI '93, ACM Press*, pp.342-347, 1993.

[20] Sears, A., Jacko, J. A. and Dubach E.M. International Aspects of World Wide Web Usability and the Role of High-End Graphical Enhancements. *International Journal of Human-Computer Interaction, 12(2)*, pp.241-261, 2000.

[21] Smith, A. & Yetim, F. (2004) Editorial: Global human-computer systems: cultural determinants of usability *Interacting with Computers*, 16 , 1-5.

[22] Taylor, D. Global Software: Developing Applications for the International Market. *Springer-Verlag, New York*, 1992

[23] Uren E, Howard R, Preinotti T. Software Internationalization and Localization: An Introduction. *Van Nostrand Reinhold*, 1993.

[24] Wilson, R. (2003) Wanted: Hispanic Professors. Chronicle of Higher Education, v50, n14, Nov 28, 2003, p. A15. Available on the web at http://chronicle.com/weekly/v50/i14/14a01501.htm (Last Accessed 09-9-2005).

[25] Yeo, W.A. Global-Software Development Life Cycle: An Exploratory Study. *Conference Proceedings on Human Factors in Computing Systems*, pp. 104-111, 2001.