IDEA GROUP PUBLISHING
701 E. Chocolate Avenue, Suite 200, Hershey PA 17033-1240, USA
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.idea-group.com
ITB10003

Chapter V

Software and Culture: Beyond the Internationalization of the Interface

Gregory E. Kersten, Concordia University, Canada

Mik A. Kersten, Xerox PARC, USA

Wojciech M. Rakowski, ANS, Bell Canada, Canada

ABSTRACT

Software applications are designed around user interaction. One interaction component is the user interface; the other deeper components represent the applications' logic and core functionality. Internationalization architectures recognize the need for localizing user interfaces to particular cultures. We continue the discussion on culture and software focusing on the software core rather than the user interface. This core corresponds to deep culture as opposed to the surface cultural manifestations embedded in the user interface. We argue here that deep culture can be embedded into application software in a modular way.

INTRODUCTION

The participants of the 1999 E-Conomy conference called for a pluralistic perspective on e-commerce and its technologies (Cioffi, 1999). Technologies developed for e-commerce have a number of popular applications, including communication and discussions, decision-making and negotiation, voting and other forms of facilitating citizens' to participate in federal and local governments. A pluralistic perspective on e-commerce and other computing technologies requires the consideration of social and organiza-

tional cultures, including value systems, beliefs and norms. The state of application software today seems to indicate that U.S. technologists and so-called, software evangelists still determine the user experience based on their own cultural biases.

To date, the development of software for a culture different from the culture of its authors has been focused on adapting the user interface. This method is referred to as *software internationalization*. The underlying assumption behind software internationalization is that all of the culturally and linguistically sensitive software components can be separated from the locale-independent core of the application (Hall, 1999, p. 298), (Nakakoji, 1996; Hall and Hudson, 1997).

The assumption of the culture-dependent interface and the culture-independent core has helped software companies to develop programs for international markets without re-writing the very same application for every new national market. The perspective that *all cultural aspects are encapsulated in the external layer of the software* has been fundamental in porting application to international markets. We will argue with this traditional engineering approach (Sommerville, 1992), which (1) separates the human interface from the mechanics of the apparatus, and (2) assumes that changing the interface is all that is necessary to change the usability of that apparatus (Laurel, 1991).

At the early stage of the innovation curve simplification and reductionism are often necessary to understand complex situations, construct models and build machines. The ability to match a problem's complexity with its representation and machinal embodiment increases with the discovery of methods and techniques that are specialized for a particular type of innovation. The discussion presented here is based on the assumption that the current state of the software development processes allows for a richer perspective on the culture-software relationship than that being employed today.

In this chapter we present arguments behind the claim that cultural concerns penetrate beyond the user interface. Contrary to what the current methods of engineering of international software would suggest, we consider *the software core being culture-dependent*. There are two interconnected motivations behind this statement: (1) the software core is a technology which, according to some theories of technology, is rooted in, and shaped by, culture (Heidegger, 1977; Feenberg, 1991; Ferre, 1995); and (2) and unlike many other technologies software describes and automates complex activities and whole processes that previously were undertaken by people and organizations. The core of a software artefact embeds decision-making, rules of behaviour and patterns of actions that depend on culture (Juustila, 1995; Kaplan, 1995; Hofstede, 1997). Consequently, embedding the attributes of the users' culture requires changes to the design of software architectures that go beyond the current international standards for software architecture and localization.

This chapter continues the discussion on culture and software and the dominant role of Western cultures, the U.S. in particular, in software development (Taylor, 1992; Juustila, 1995; Kaplan, 1995; Nakakoji, 1996; Carmel, 1997; Kersten, Matwin et al., 2000). Two perspectives on culture are presented, followed by three theories of technology. Then perspectives on culture are coupled with theories of technology. This theory-based assessment is illustrated with three examples of culture being embodied in the software core. A proposal for software "culturalization" is then formulated. It is based on the recognition that deep cultures can be embedded in software. This should happen, as we conclude, even if one assumes that the impact of national cultures diminishes.