Chapter 10
Representing Culture via Agile Collaboration

Craig Dietrich
University of Southern California, USA

John Bell
University of Maine, USA

ABSTRACT

Creating software that supports cultural knowledge management brings developers face to face with issues they may not encounter when dealing with more general-purpose applications. Many times cultural specialists will have a unique understanding of the data, relationships, and special sensitivities that should be reflected in the interface and structure of software intended for use in a specific field. When general-purpose software is not able to accurately capture these subtleties of culture, experts and developers can work together to create small, focused solutions. This chapter discusses the special issues presented when developing software for cultural or creative organizations, the development philosophy behind targeted applications, and methods to design ecosystems of small applications that can be combined to meet a wide variety of needs.

INTRODUCTION

Freely available Web-based publishing tools such as *WordPress* have made mid-level Web development considerably easier than previous write-your-own methods. Certainly, *WordPress*'s well-advertised "5 minute install" (WordPress.org, 2009) makes one wonder why all computer-related projects couldn't be so easy. Yet the smallest changes to existing systems such as *WordPress* create complex challenges. Adapting HTML, rearranging categories, or customizing publishing templates are changes that can be made superficially. Adding logic based on cultural parameters such as family affiliation, sacred status, or other requests require re-engineering. Culturally-sensitive software projects need not only a plan to manage content but also a complex set of protocols to represent culture. These complexities require technical innovation beyond what is possible with many general-purpose tools.

DOI: 10.4018/978-1-60960-044-0.ch010

The challenge faced by developers is that cultural protocols are rarely fully definable. Asking a programming or design team to create an application based on a bullet-point feature list or occasional conversation with experts is not likely to result in an application that reflects the culture's nuances. Developers are trained to build software that is generalized and reusable because it is efficient, but what is often lost is the flexibility to incorporate new or outside-the-mainstream concepts. When a cultural specialist requires specialized software, other methods may be sought out to meet both the developer's goals for efficiency and the expert's need of flexibility. One possible method of reconciling these superficially contrary goals is to build loosely interconnected specialized tools.

ISSUES OF REPRESENTING CULTURE

Ready-to-use content management systems such as *WordPress* feature user-friendly sets of administration pages and publishing templates for quickly deploying blogs. While *WordPress* includes widgets and plug-ins to extend its functionality, collectively they serve to annotate blog posts, not cultural paradigms. A blog is a serialization of linear thought: today I have an idea, tomorrow another. The structure imposed by the software turns life into a sequence of events: drove to work, met some deadlines, saw something interesting, and came home. If this oversimplification of daily routine does not sit well (certainly, life is more complicated than a series of events) then publishing tools might need to push their paradigms further to capture the breadth of human experience.

Many developers of large-scale software do not interact sufficiently with the groups that use their products and therefore misjudge the needs of culturally-sensitive projects. When users have needs that require new functionality, they send feature requests to the off-site development team. Some teams see this separation as an advantageous business concept, allowing conformity and parallel development of features within the same project. For this reason, it is a model used by many industries, including retail where large chain stores stock standardized inventories of products manufactured in industrial centers. Bottlenecking goods and profits is great for efficiency but marginalizes those that might be seeking to use systems for purposes outside the mainstream. When representing culture, it's a society's practices–not profits–that risk being bottlenecked.

Though business analysts might appreciate the efficiency of bottlenecks, for creators of culturally-sensitive software bottlenecks threaten to squeeze out the same nuances they are hoping to represent. Many Indigenous communities, for example, see gender, sacred status, and family affiliation as important protocols to be included in digital archives. (Christen, 2008) Unfortunately, there are few existing archive systems that include these protocols in their sorting and access algorithms.

In the face of oppressive digital rights management (DRM) imposed by media corporations (Fisher, 2007)–often seen as burdensome by consumers (Pepper, 2008)–further restricting access to content based on gender, sacred status, or family affiliation may seem like something to argue against. This issue was recently debated on technology hub *Slashdot.org* after the site posted a description of the *Mukurtu Archive.* The archive was built in collaboration with the Indigenous Warrumungu community in Western Australia and embeds the community's protocols and restrictions directly into the interface. A *Slashdot* commenter argues, "The users actually *want* the rules to be enforced on them. It's more to protect them against *accidentally* viewing stuff that they're not supposed to while searching for other documents" (ArsenneLupin, 2008). While useful for the members of the archive's own community, there are also implications for cross-cultural communication described by another *Slashdot* commenter:

Related Content