

Chapter 1.10

Free/Libre Open Source Software for Bridging the Digital Divide

Yu-Wei Lin

University of York, UK

INTRODUCTION: THE FORGOTTEN USERS IN SOFTWARE DESIGN

As some scholars claim, the digital divide, referring to the perceived gap between those who have access to the latest information technologies and those who do not, entails that not having access to this information is an economic and social handicap (Compaine, 2001). In software design, structured inequalities operate along the main axes of gender, race/ethnicity and class. Each of these in turn generates its own structure of unequal practices giving rise to institutionalised sexism, racism or class divisions/conflict. “Gender, race and class also crosscut each other in various complex ways, sometimes reinforcing and at other times weakening the impact of existing inequalities” (Cohen & Kennedy, 2000, p. 100). For instance, Webster’s research (1996) employing feminist approaches to study computer system designs addresses the issue of a male-dominated system design field, which continuously excludes female users’ needs, requirements, interests and values in the innovation process. She criticises that, “Human factors may be bolted onto existing

methods of systems design, local and contingent knowledge of work and information handling processes held by users in an amorphous sense may now even be incorporated into the systems design process, but this does not create an awareness of the way in which skills and knowledge are defined in gender-divided terms” (p. 150).

In a similar course, I argue that users’ experiences in developing or undeveloped countries are often ignored in mundane software designs led by developed countries. Although localisation of information infrastructure is an eminent issue emerging in current system development, profit-oriented products and services, such as Microsoft’s local language program (LLP), do not really comply with local needs. Rather, this type of multi-languages software packages, a software suite fabricated universally for countries around the world, signify the phenomenon, which I term the “MacDonaldisation of Windows-Intel platforms,” which in fact alienates users and the local contexts.

FLOSS IN ACTION

In recent years, free/libre open source software (FLOSS) has emerged as an important phenomenon in the information communication technology (ICT) sector as well as in the wider public domain. An increasing number of governments have endeavoured to either convert the public administration infrastructure from Windows to Linux or to adopt FLOSS for similar tasks (e.g., Munich in Germany or Zaragoza in Spain) (c.f., “Linux in Spain” on LWN.net; C|Net News.com August 29, 2001). FLOSS transparentises the often black-boxed software code and allows users to copy, distribute and modify a programme received freely. In making source code available, software technologies can be challenged, adapted, and ameliorated to satisfy diverse user needs. Apart from solving the prolonged usability¹ problem in software engineering, implementing FLOSS also helps ground both social and technical knowledge in locales and bridge the digital divide. In other words, implementing FLOSS facilitates technical knowledge (e.g., programming skills and ICT expertise) and social experiences to be transported and transferred through the acclaimed practices of social networking and mutual help noted prominently in many recent community studies (e.g., Wellman, 1999; Rheingold, 2000; Hampton & Wellman, 2003; Jordan et al., 2003; Lin, 2004a).

There have been a number of tactical considerations of implementing FLOSS in countries or organisations devoid of intellectual or financial resources: for economic reasons to save software costs; for educational reasons to improve human resource; for political reasons to stop monopoly proprietary software from expanding their market share as well as to gain digital autonomy, just to name a few. Hence, it is a strategic interplay for local governmental or non-governmental organisations (NGOs), and FLOSS activists to coalesce to tackle these inequalities. Because knowledge transfer is as crucial as infrastructure

implementation, hands-on training made available to the local users is essential in the execution. Projects such as the E-Riders² and Low Income Networking and Communications³, or events such as the Summer Source Camp⁴ and Africa Source⁵, all illustrate the transfer of knowledge and technology across cultural borders. These examples also show how the implementation of FLOSS shapes the lives and identities of local users as well as software developers around the globe (Lin, 2004b). Additionally, there is conspicuous implementation of Linux-based infrastructure in the local educational, NGO and governmental organisations in developing countries or regions (e.g., Washington Post, November 3, 2002). The advents of embedded technologies such as the “Simputer,” a Linux handheld applied in India, are believed to enable affordable, sustainable village development in places without phones and power, giving more and more people a voice in the conversation about their future (Cherlin, 2002). Wireless technologies are amongst others to bring the Internet to developing countries or regions to facilitate networking at both local and global levels. Krag, a Danish expert of wireless technologies who I met at the 2003 summer source camp in Croatia, describes wireless technologies as low-cost and decentralised. Here is a quote from his talk at the O’reilly 2004 emerging technology conference⁶ about the advantage of wireless technologies:

Billions of people in the world have never been online. The Internet as a technology is an elitist tool, reserved for the few and unreachable by the many. This is a problem not likely to be solved by the commercial interests of existing telecommunications companies and existing ideas about expensive, centralized infrastructure. But low-cost, decentralized wireless technologies could have an important role to play, in this respect. Their low price point and decentralized nature, and the openness of the standards, mean that these technologies are incredibly adaptable to new situations and new uses. (Krag, 2004)

5 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

www.igi-global.com/chapter/free-libre-open-source-software/29382

Related Content

Improving Effectiveness of Process Model Matchers Using Wordnet Glosses

Mostefai Abdelkader (2019). *International Journal of Systems and Software Security and Protection* (pp. 1-19).

www.irma-international.org/article/improving-effectiveness-of-process-model-matchers-using-wordnet-glosses/247489

Information Feedback Based Architecture for Handling the Scalability Issues in the Reusable Cloud Components

Manjunath Ramachandraand Pandit Pattabhirama (2012). *Software Reuse in the Emerging Cloud Computing Era* (pp. 186-202).

www.irma-international.org/chapter/information-feedback-based-architecture-handling/65172

Where to Integrate Security Practices on DevOps Platform

Hasan Yasarand Kiriakos Kontostathis (2016). *International Journal of Secure Software Engineering* (pp. 39-50).

www.irma-international.org/article/where-to-integrate-security-practices-on-devops-platform/176400

Modelling Security Using Trust Based Concepts

Michalis Pavlidis, Haralambos Mouratidisand Shareeful Islam (2012). *International Journal of Secure Software Engineering* (pp. 36-53).

www.irma-international.org/article/modelling-security-using-trust-based/66407

The Method and Tools Development for Web-Cameras Color Correction in Binocular Vision Systems

Konstantin Dergachov, Leonid Krasnov, Olexander Cheliadinand Olexander Plakhotnyi (2021). *Handbook of Research on Software Quality Innovation in Interactive Systems* (pp. 154-191).

www.irma-international.org/chapter/the-method-and-tools-development-for-web-cameras-color-correction-in-binocular-vision-systems/273569