Babel 2.0: Bridging the Linguistic Digital Divide

By: Brad Koegler

Brad is a Master of Information (MI) candidate at the University of Toronto. He hails from Waterloo, Ontario, Canada and holds a Bachelor of Arts (Honours) in Sociology from the University of Guelph. He has worked as a bookseller, a residence assistant, and a music journalist. His areas of interest include language and translation, cyberculture, social effects of new media, and the information economy. Plans for the near future include writing a Master's thesis on the linguistic digital divide, then pursuing a doctorate in Information Studies.

brad.koegler@utoronto.ca

Abstract

The Internet's bias toward the English language is not surprising considering its origins as an American military network, but it is a highly problematic access barrier for an ostensibly global communications medium. Information and communication technologies (ICTs) in general tend to favour English, creating a "linguistic digital divide" wherein even those with access to technology find difficulties expressing themselves in their native languages. The ASCII character standard creates difficulties for those using non-Latin characters, and solutions from standards organizations such as ICANN have been both slow and piecemeal. Proprietary software makers have similarly lagged in localization efforts and are only now beginning to catch up to open-source projects. Machine translation has advanced rapidly, but is still not a viable solution. In the meantime, English's incidental supremacy makes it an ideal Internet lingua franca, but increased accessibility in other native languages will benefit all.

"La parole humaine est comme un chaudron fêlé où nous battons des mélodies à faire danser les ours, quand on voudrait attendrir les étoiles." (Language is a cracked kettle on which we beat out tunes for bears to dance to, while all the time we long to move the stars to pity)

-Gustave Flaubert, Madame Bovary
Introduction

The rise of the information society and its backbone, the Internet, has been startling in its speed and scope. The military programmers who set up the first ARPANET packet-switching network in 1968 could hardly have imagined its present size and ubiquity; in developed nations, it is now almost difficult to participate fully in society without Internet access, and even developing countries are seeing notable increases in the use of information and communication technologies (ICTs). In 2009, Spain and Sweden went so far as to declare broadband Internet access to be an inalienable right for their citizens (Stross, 2009, ¶ 4). The abundance of such technology has meant democratization of the means of production in what Benkler (2009) calls the networked information economy, “an economy of information, knowledge, and culture that flow through society over a ubiquitous, decentralized network” (p. 1246).

A dominant concept in the discourse surrounding this paradigm shift has been the idea of the “digital divide,” a nebulously-defined problematization of the gap between those who use ICTs and those who cannot or will not. Through government and corporate initiatives – whether in terms of universal service like in Spain and Sweden or universal access like the Bill Gates-fuelled public library project in the United States – great strides have been made toward bridging the digital divide, at least as it is defined in the discourse of the neoliberal state (Stevenson, 2009, p. 18). However, bringing the world to the Internet has revealed another digital divide, one which “mirrors traditional patterns” of division and oppression (Hindman, 2009, p. 9). Even once citizens begin to use computers and the Internet, the same old divisions and hierarchies remain. Nowhere is this more true than in terms of language. The very infrastructure upon which the Internet is built is profoundly biased in favour of English content, at the expense of users wishing to communicate in other languages. This paper will examine the two most common manifestations of the linguistic digital divide: native-language support in software, and “ASCII imperialism” on the Internet (Pargman & Palme, 2009, p. 197).

Linguistic Imperialism or Lingua Franca?

When unpacking and analyzing the linguistic digital divide, it is important to stress the difference between linguistic imperialism and the rise of a lingua franca. Discourse about the ascendancy of English often carries a cynical, suspicious tone, but Anglophone supremacy in ICTs came about more as a matter of convenience than one of wilful oppression.

There are several possible outcomes when groups who do not share a mother tongue attempt to communicate. If one group is dominant, the other might – whether willingly or coerced – learn the more powerful group’s language. Two groups of more or less equal standing without any languages in common might craft a “pidgin” language that blends the two mother tongues. However, if the groups are
lucky enough to share a common language, they will most often use that *lingua franca* (from the Latin for “Frankish language,” in reference to an Italian-based pidgin used by traders and diplomats in medieval times) to communicate (Chirikba, 2008, p. 31).

English has been a global *lingua franca* for upwards of a century. The supremacy of the British Empire, then the dominance of the United States, spread the use of English all around the world. It is one of six official languages of the United Nations (along with Arabic, Chinese, Russian, Spanish, and French), and one of three working languages (along with French and German) of the European Union (UN DGACM, 2002, ¶ 1). It is perhaps no surprise, then, that English has also become the *lingua franca* of the Internet. Native speakers of English only make up twenty-nine percent of the wired world, and this number is dropping rapidly – Chinese speakers will have overtaken English by the end of the next decade – but the content of the Internet remains four-fifths English (Currie, 2007, ¶ 3).

Despite reflexive hand-wringing about yet another expression of Western dominance, having a common language on the Internet actually serves to benefit its users. The Ethnologue language database counts 6,909 languages currently in use (Lewis, 2009, Table 1); to resist the use of a *lingua franca* would reduce the Internet to a new Babel. English’s pre-existing ubiquity, however ill-gotten, makes it the logical choice. Allegations of linguistic imperialism become more serious, however, when the users who use English to communicate with other groups cannot use their own languages to communicate amongst themselves.

**Whither Localization?**

Musinguzi (2008) discusses Kiganira Kijambu, a Ugandan entrepreneur whose biography is a techno-utopian NGO’s dream: using an ICT training centre near his home in Mayuge District, he was able to receive training as an accountant and use e-commerce technology to triple his farm’s profits (¶ 2). Mr Kijambu’s accomplishments are impressive, but he had a decisive advantage: unlike all but ten percent of his fellow Africans, he speaks English (Musinguzi, ¶ 6).

Mr Kijambu’s mother tongue, LuSoga, is not supported by any of the major computer operating systems, and is all but absent from the Internet. This creates a vicious circle against the implementation of ICTs: lack of utility leads to lack of demand, which leads to a lack of incentive to develop software and content for LuGosa, which only reinforces the lack of demand by LuGosa speakers for computers with which they cannot use to effectively interact. While online business has allowed Mr Kijambu to increase his business threefold (Musinguzi, 2008, ¶ 3), broader language support would not only further improve his business, but would also allow his non-Anglophone peers the same kind of opportunity.

Keniston (1997) defines “localization” as “the translation of programs originally written in and for one language into intelligible and
user-friendly versions in and for another language” (¶ 14). In the early days of programming, this was done as an “add-on,” if at all; software would be coded entirely in English and, if unavoidable, a localizer might be hired to translate the menus and manual into Spanish or Japanese. The culture of early computing was such that Linus Torvalds, a Swedish-speaking Finn and the creator of Linux, did and still does all of his code comments and annotations in English for simplicity’s sake (Torvalds, 1997, ¶ 1). In effect, the Anglophone “technology capitals” of Redmond, WA and Cupertino, CA were dictating the information policy of the world at large.

Now, however, to only release a program in one language is to greatly diminish its appeal. Even Microsoft, historically a laggard in terms of localization, has greatly broadened its language support; Windows XP supports over 140 languages, and the nascent Windows 7 currently supports thirty-eight, with plans to support “more than ever before” within the next year (Bennett & MacConnell, 2009, ¶ 34).

FLOSS (Free/Libre Open Source Software) tends to be more nimble in this regard. The source code for proprietary software is rigid and uniform; adding support for other languages later requires jury-rigged service packs and add-ons that often cause as many problems as they solve. Anyone who has tried to work with foreign-language translations of American programs – be they operating systems or applications – can attest that the problems defined here as “technical localization” are far from universally solved. Absurdities abound; programs that operate well in one language, with one set of characters, crash in another; non-English speaking users may be presented with unintelligible, awkward, or English-language-only help files, tutorials, or documentation; and so on (Keniston, 1997, ¶ 16).

Meanwhile, open-source localization, even in an established system, is as simple as editing and recompiling the code. The greatest strength of FLOSS is the “L” for “libre,” what Stallman (1993) defines as “free as in speech, not as in beer” (¶ 2). According to the Free Software Foundation’s definition of free software:

You should also have the freedom to make modifications and use them privately in your own work or play, without even mentioning that they exist...In this freedom, it is the user’s purpose that matters, not the developer’s purpose; you as a user are free to run a program for your purposes, and if you distribute it to someone else, she is then free to run it for her purposes, but you are not entitled to impose your purposes on her...Freedom [one] includes the freedom to use your changed version in place of the original” (Free Software Foundation, 2009, ¶ 6-10).

The autonomous region of Extremadura, Spain has been highly successful in using ICTs to improve their economic standing, and in using FLOSS to work around the linguistic digital divide they might have otherwise faced. Extremadura was the poorest region in Spain, predominantly agricultural and facing an exodus of its youth to more prosperous and/or urban regions. In 1997, the region’s president launched a “Regional Strategy on Information Society”
that examined how ICTs might be deployed to achieve “accessibility for all; Internet as a public service; and stimulation of technological literacy” (Van Disseldorp, 2008, ¶ 6). With the help of funding from the European Commission, the Extremaduran government launched eExtremadura, a regional intranet with three major components: “knowledge centres” for promoting information literacy; a technical network linking schools, hospitals, and other public buildings; and Vivernet, a “business incubator” designed to encourage entrepreneurship (Van Disseldorp, ¶ 8-11).

This project would have been impossible, or at the very least prohibitively expensive, were it not for the use of open source software. Rather than approaching a proprietary software firm, the Spanish government hired programmers to download a copy of the Debian distribution of Linux and customize it for the needs of the Extremaduran people (Disseldorp, 2008, ¶ 16). The resulting operating system, gnuLinEx, was customized far beyond what would have been possible with closed software: it is localized in Spanish and various other Extremaduran dialects (many Extremadurans speak uncommon dialects of Portuguese, or the endangered Extremaduran language), and the software is all specially configured in conjunction with local laws and standards. Even the programs have names particular to Extremaduran culture; the word processor, for example, is named for the famed local poet Espronceda (Disseldorp, ¶ 21).

Perhaps most attractive, though, was the price. The entire software project only cost €193 562.84 for both development and deployment, while the Spanish Ministry of Education, Science, and Technology estimates that a comparable proprietary solution would have cost €30 million, and still might not have had the same degree of customization as gnuLinEx (Disseldorp, ¶ 24). As a result of eExtremadura, the economic gap between Extremadura and the rest of Spain has narrowed, the “brain drain” has slowed, the local quality of education and healthcare has improved, and many new businesses and industries are starting up (Ghosh & Schmidt, 2006, p. 4).

Ghosh and Schmidt (2006) are especially adamant about FLOSS as a tool for bridging divides:

'[The] link between open source and the rise of small ICT businesses is especially important given the tendency of proprietary vendors to ignore local needs, especially in developing regions...Developing countries need to avoid being locked out of acquiring skills and competencies. The adoption of open source policies provides environments that promote skills development and the ability to create (p. 3).

An initiative like eExtremadura might not necessarily be exactly the right one for other developing regions like Mayuge District, Uganda, but it is difficult to argue that – for both cultural and economic reasons – any sort of ICT solution should not be open source.
**ARPANET, ICANN, and ASCII**

The near-anarchic democracy of today's Internet belies its origins as a tightly secured tool of the establishment. Before there was an Internet, there was ARPANET, a packet-switching network connecting military computer databases in the United States. In those early days, little thought was given to the idea of needing to use other languages, so the architects of ARPANET did not bother programming foreign-language support. The international standard for printing characters, ISO 646, had various international versions that allowed for some non-English characters, but it was the American variant (American Standard Code for Information Interchange, or ASCII) that became the backbone of the Internet (Pargman & Palme, 2009, p. 183).

Nisselbaum and Friedman (1997) enumerate three different categories of bias in computer systems: inherent, technical, and emergent (p. 25). The use of the ASCII standard is a case of the third; the architects of the Internet were designing a national communication system, not an international one, and did not anticipate the later need to accommodate all of the world's languages. The problem, however, has become an incredibly difficult one to fix. Problems with character standards are as old as the printed word. German characters bearing umlauts, for example (ä, ö, ü), were uncommon on typewriters and telegraph keyboards outside of Germany, so Germans developed a convention of replacing the umlaut with an “e” after the bereft letter (e.g. the author of this paper is “Koegler” in Canada but “Kögler” in Germany). However, such solutions are not always possible, feasible, or even desirable. German only needs to replace the umlaut to conform to ASCII. Chinese speakers would have to discard their writing system altogether.

Of course, anyone who has used the Internet recently is well aware that websites can display non-ASCII characters ranging from Chinese and Cyrillic characters to smiley faces and hearts. The left sidebar of Wikipedia is full of links to other languages' Wikipedias, rendered in their own scripts. However, all of this is artifice layered on top of the same ninety-five ASCII characters that the Internet began with. Fonts like Unicode can support millions of characters, more than in all the world's scripts, but the information travelling between computers must first be translated in and out of ASCII. The German spelling of my name, typed into an e-mail in HTML, would be transmitted as “K&ouml;gler.” This seems harmless enough at first. Ghosh and Schmidt (2006) discuss the idea of a “natural monopoly,” a technical standard that, while still monopolistic, is seen as a lesser evil because it increases network externality (p. 3). We accept restrictions and regulations for things like rail gauge and wireless bandwidth because we value the ability to use our devices anywhere, prioritizing “how well it interworks with other stuff [rather] than its own intrinsic merits” (Levien, 1998, ¶ 4) It is difficult to imagine a functioning global Internet without a similarly strict transfer standard, but ASCII
simply does not handle foreign languages well. Pargman and Palme (2009) cite a study wherein the translation company WorldLingo sent customer service e-mails written in Japanese script to forty Fortune 500 companies (p. 179). Only one replied in Japanese; most didn’t reply at all. The remainder of the few who did reply said that the messages they received were “corrupted.” These ostensibly international conglomerates did not have support for Japanese characters on their customer-service department computers and were baffled by the raw code they saw instead.

The technology to fix this exists. The Internet Corporation for Assigning Names and Numbers (ICANN) and World Wide Web Consortium (W3C) could change the standards and begin using Unicode, allowing speakers of all languages to interact with the Internet directly instead of through the artifice of ASCII. It would be a costly process, and problematic for the oldest computers, but the long-term benefits on the world stage would far outweigh the short-term troubles. Unfortunately, these “arcane standards boards,” once bastions of techno-utopians and open-source activists, have been co-opted by “an influx of people from large commercial companies” (Pargman & Palme, 2009, p. 192). The inherent conservatism of committees, combined with reluctance on the part of the Microsots and Googles of the world to have to adapt their software, means that an epochal shift to correct the emergent bias is highly unlikely.

Small concessions have been made. On October 30th, 2009, ICANN announced the Internationalized Domain Name Fast-Track Process, wherein they will now accept domain names that contain non-ASCII characters. Despite all the fanfare, this was too little, too late. The international domain names (IDNs) are encoded at the user level using a system called Punycode, but most of the aforementioned problems still apply: the Punycode will still be sent out to the Internet as ASCII, with the hope that the computer at the other end will be able to convert it back (Klensin, 2004, p. 2). It is perhaps telling of ICANN’s level of commitment to internationalization that, at the time of this writing, the press release for the IDN Fast-Track Process (2009) is still only offered in English.

**Conclusion**

As with many other facets of the Internet, Google has been a loud and powerful presence in the field of computer translation. Their Google Translate service uses a twenty-billion-word corpus of public-domain United Nations documents, statistical analysis, and crowd-sourced corrections in an attempt to provide instantaneous translation of any page on the Internet (Tanner, 2007, ¶ 2). It is arguably the best such service available right now, but it still has a long way to go before it can be used for much more than approximations.

In the meantime, there will be a place for English as the lingua franca of the Internet. Its current ubiquity makes it likely that it will maintain this position for quite some time. However, ICTs in general and the Internet in
particular must expand support for other languages. All of the rhetoric about ICTs as new cornerstones of democracy is useless as long as there are people who cannot use those technologies in their own languages.
References


