



Software colonisation

There is a worrying dimension to the uptake of technology in Africa, which is perhaps most vividly illustrated where I live in South Africa, says *Jarred Cinman*. As sub-Saharan Africa's wealthiest and most technologically-advanced country, South Africa is a good indicator of how Africa is embracing computing and other technologies. The worrying part is that South Africa is a massive net importer of key technologies, and in particular here I refer to software. An owner of a local software development business myself, it is shamefully obvious that despite the benefits accrued through the local utilisation of software, vast sums of money are flowing offshore back into

➡ US and European hands.

I would contend that this is yet another example of Western colonialism, the key ingredient of which is keeping us as “a market” for Western products, instead of encouraging our own intellectual growth, and building our own assets.

In the software world this is already a complex story, despite the relative recent arrival of the Internet age. First, there's the whole issue of what's being called Floss (Free/Libre/Open Source Software). Many developing countries (Brazil is a particularly good example), have embraced Floss with open arms, and the South African government has followed suit by adopting policies which promote the use of Floss at least as an equal to commercial software.

The key principle behind Floss is its greatest strength and its greatest weakness, however. And that is the idea that software somehow intrinsically belongs to all people, propounded by Richard Stallman, “father” of free software, and that keeping the source code private or proprietary is in some sense “wrong”. This quickly breaks down into a discussion of the invalidity of “intellectual property” full stop, which argues that both knowledge and software should be free and not respect national borders.

The emotional appeal of this approach is undeniable. And it doesn't stop there. Economically it makes fantastic sense for developing countries to have access to “free” software, and to be able to implement top-notch technology at a fraction of the cost of the proprietary alternatives. And of course there are the arguments about software quality and community development which fall outside the scope of this article.

The problem, I believe, is that African proponents of Floss are glossing over an important subtlety in the rush to be at one with peace, love and the software world. And that is that despite their Stallmanesque views of intellectual property as applied to software, national boundaries do exist in the real world. Software exists within the same global economy and socio-political dynamics that any other kind of property does – be it cash, commodities or equities.

Intellectual property is not just a matter of who owns what, it's also a matter of who has the skills and ability to make, invent and control. Linux is open source, but in fact Linus Torvalds still holds a pre-eminent role in deciding what makes it into the kernel and what doesn't. And Linus is a Fin who lives in California.

Africa, despite benefiting enormously from Linux – don't misunderstand me – is still for the most part a *user* of this technology, not a creator or controller. As much as the free software people want to argue that ownership is irrelevant in their world, my argument is that it's definitely not.

Here are some reasons why:

- Because national borders exist, interests are still largely national. It's easy to talk of a global village, but witness the recent actions of the United States toward Iraq in the name of protecting its national security. When it comes down to the crunch, countries will still look out for themselves first.
- While nowhere near as serious in its implications, software betrays national interests too – whether consciously or not. What a piece of software does and doesn't do, what the vision is, even which projects are selected, are defined by needs in the country of origin. To the extent that open source draws in developers from other countries, they are for the large part other First-World countries.
- Even if “intellectual property” is a swear word, intellect, skills, talent and ability are not. By supporting open source projects based in the First World, even just by opting for Drupal or Debian, instead of Cambrient Contentsuite or Ubuntu, Africans are contributing to first world dominance, encouraging

“Witnessing the recent actions of the United States toward Iraq in the name of protecting it's national security should be a loud wake-up call to all of us. When it comes down to the crunch, countries will still look out for themselves first.”

more funding of these projects, and the growth of skills and opportunities of developers over there.

Africa's ability to become a technological superpower in its own right is diminished. Our base of genius hackers, our suite of home-grown software remains tiny, and – here's the colonialism – we are pulled along into the Western way of operating.

Here's a hypothetical, though not much of a stretch, example. My business is a content management software developer based in Johannesburg. Content management systems are a dime-a-dozen, and there are several good Floss ones out there. There are also many expensive and proprietary systems from the likes of Microsoft, IBM, Vignette etc.

Let's say an African corporate or government decides to undertake a content management project, and implement a CMS. They have the following broad options:

1. Purchase a US or European product, probably through a localised reseller.
2. Build their own system.
3. Implement the project on a Floss CMS, once again, US- or European-“owned” (where it was made and controlled).
4. Purchase a local proprietary CMS.
5. Utilise a local Floss CMS.

Right now, the reality is that most big corporates in Africa – certainly in South Africa – are making the first choice, and occasionally the second. Government policy in South Africa, Namibia and others means they may opt for choice three. A few are showing faith in local software, and there are no Floss CMS systems based in Africa that I know of.

CMS is just a familiar example to me, but the same is true in many, even most, realms of software. Africa is an importer, a user, an implementer.

Why is that a problem?

- For proprietary systems, cost first and foremost – in terms of direct US dollar or Euro licensing fees, usually with annualised upgrades, as well as the importing of consultants and other skilled professionals to undertake implementations (experience has shown this is just about always required).
- Local ways of doing things are sacrificed, diluted and made subservient in the interests of a bigger “generic”, First-World way of doing things.
- Even where Floss systems are selected, little or no intellectual growth takes place in Africa – largely we take offshore systems and implement them, with some tweaks.
- Additional economic growth as represented by a thriving African software development community – either through the purchase of locally developed software, or the funding of local Floss projects – is missed out on.

Now, the African proponents of Floss would argue that I'm missing the underlying principles of Floss to argue for a software Africanism. They would also argue, perhaps validly, that my argument would lead to reinvention of the wheel simply for the sake of having an African wheel. Why not use the best-of-breed system no matter where it's from? The only issue is freedom in the “libre” sense of the word.

They would also argue that encouraging a lo-

cal proprietary software market is to take one step forward and two steps back.

But this allegiance between the developing world and open source has to be carefully evaluated. I have no doubt that open source people – the Free Software Foundation, the Linux movement and Ubuntu, to name a few – have good intentions. And I have no doubt that they would like to see the developing world uplifted and national borders dissolved.

And I also have no doubt that there are certain types of software – say operating systems – which it would be foolish to write again. Linux probably is about as universal as we need it to be right now, and work by the likes of Ubuntu with translate.org.za makes it suitably localised.

But I do ardently believe that Africa needs to get African about software. That means two things:

First, there should be strong local quotas for local software. This is already in place with content such as radio and television in South Africa, and is stimulating the film and music industries enormously. And yes, this means proprietary software for now.

This is a weakness in the Floss argument as proposed by Africans: they would prefer a US-based Floss system to an investment in an African proprietary system. I say that's just wrong. Even tactically, what we want is to encourage local skills and talent so that we can meaningfully participate in Floss projects. And – most importantly – that we can start spending our software licensing money here.

The fact is Africans are going to be buying software, now and for the foreseeable future. Businesses, in particular, are conservative, and the Open Source message has been poorly tailored to talk to boards of directors. The way corporate spending works, purchasing software is a capital expense, and that's an easy concept to grasp. Getting something for free, paradoxically, is often impossible for a corporation to understand.

Second, there should be passionate funding of African Floss projects. Whether an individual software developer could be encouraged to open source their code in the pursuit of the great free software ideal, or whether new projects have to be identified and initiated, governments and funders should be getting behind African-initiated projects.

I started by saying that Africans being users and implementers of offshore software is a kind of new colonialism. Partly it's draining of cash – in the case of South Africa, something like R8-billion a year in offshore software licensing according to Nhlanhla Mabaso from the Meraka Institute at the recent Go OpenSource conference. Partly it's the stunting of African software development.

This last is a subtle point in some kinds of software, and a lot less subtle in others. Sometimes it's about language, but sometimes it's about a less definable “African-ness” of experience, culture and process, which cannot just be dismissed out of hand.

If this African-ness is not built into software applications where it matters, and if we continue presuming that how a system works in the US and Europe is how it should work here, then we are in fact being colonised in a very real way.

Even if the intentions start far from that place. ■

