# Free/Libre & Open Source Software and Open Standards in South Africa

## A Critical Issue for Addressing the Digital Divide

National Advisory Council on Innovation
Open Software Working Group

July 2004

Version 2.6.9

# Version One

Contributing team and their affiliations as of 31/1/2002

**NACI Coordinator:**
Sibusiso Sibisi (NACI)

**Main Author:**
Sibusiso Sibisi (NACI & UCT)

**Major contributions:**
Mike Jensen (Consultant)
Philip Machanick (Univ. of Witwatersrand - Computer Science)
Edwin Blake (Univ. of Cape Town - Computer Science)
Evan B. Summers (Obsidian Systems)
Dwayne Bailey (Obsidian Systems)

**Additional input:**
Peter Underwood (UCT - Centre for Information & Library Studies)
Johan Eksteen (CSIR)
Luci Abrahams (NACI)
Titus T. Singo (NACI)

# Version Two

Contributing team and their affiliations as of 30/4/2004

**NACI Coordinator:**
Sibusiso Sibisi (NACI & CSIR)

**Main Authors:**
Thomas E. Pogue (CSIR - Policy Group)
Bob Day (Consultant)

**Major contributions:**
Adi Attar (CSIR/ – Open Source Centre )
Nhlanhla Mabaso (CSIR – Open Source Centre )
Sibusiso Sibisi (NACI & CSIR)

**Additional input:**
Errol Epstein (Nandos')
Charles Irwin (Lewis Stores)
Robert Jolliffe (UNISA - Computer Science)
Rasigan Maharaj (CSIR - Policy Group)
Thabo Mogaswa (Kgatelopele Technologies)
Alastair Otter (Tectonic Media)
Aslam Raffee (GCIS)
Sean White (SITA)
Kim Tucker (CSIR – Open Source Centre)
Neo Maledu (CSIR – Open Source Centre)

## Executive Summary

A number of changes have occurred in the South African FLOSS[1] landscape since the first version of this document was produced in 2002[2]. This revision of the original document extends the discussion of a range of issues, both local and global, that continue to have impact on the progress and potential of FLOSS. These discussions take advantage of the increased research into the FLOSS phenomenon which has occurred over the past two years. The document also indicates the significant progress that has been made since the first version in making FLOSS adoption a priority in South Africa's public sector.

Few would dispute that the Internet is having a dramatic impact on most aspects of human society in the developed and in the developing world. Terms such as the Information Society, the Knowledge Economy, the Digital Village and the Digital Divide dominate discussions at all levels of the connected society whilst amplifying the exclusion of the unconnected. Behind all these issues lies the Internet. Yet, 13 short years ago, the Internet was far from being an accepted technology or phenomenon, and many, especially in the IT industry multinationals at that time, dismissed it as 'yet another fad'. The FLOSS phenomenon is clearly making its presence felt in the global networked world. A third mode of production, "commons-based peer-production", is emerging, beyond property and contract[3], with both ethical and practical implications for sustainable development world wide.

But what should the rest of South African (and African) Society be doing in terms of FLOSS and open standards in order to be an active part of the FLOSS phenomenon? This document tackles this question in relation to the following sectors:

- *Large Private Sector Organisations:* Many large South African organisations have been employing ICTs for years. There are examples of companies successfully implementing FLOSS solutions since the mid 1990s. We recommend that every large organisation should, at the very least, be objectively considering the benefits they would experience if they were to migrate to FLOSS in the near future.
- *Small Private Sector (SMMEs):* ICTs offer great possibilities for improving SMME business practices, productivity and market reach. On account of the lower entry level offered by FLOSS, many more could experience the benefits, and via a multiplier effect, grow the national economy. a truly *indigenous ICT industry will emerge with opportunities for* SMMEs to provide FLOSS-based installation, customisation, training and support services.
- *Civil Society:* A wide range of initiatives are currently attempting to provide ICT access, literacy and connectivity to the poorest communities (from the public sector, the international development community, NGOs, etc.). Although the information about FLOSS is freely available, and funders are starting to

[1] Free/Libre and Open Source Software. See http://www.gnu.org for background on Free Software and APPENDIX ONE: FLOSS Definitions.
[2] See: http://www.naci.org.za/
[3] See: http://www.benkler.org/CoasesPenguin.html

recognise its potential, widespread impact is yet to be observed. Strong advocacy to decision makers and stakeholders is recommended for FLOSS to become an essential component of all ICT for development initiatives in order to effect such impact.

- *Environment and Agriculture:* FLOSS is already playing a significant role in several tools used by these two, related sectors which are so important in the sustainable development of South Africa and Africa. However, there is a need for communication and coordination to optimise use of FLOSS to serve mutual needs in these sectors.
- *The International Development Community:* Some global organisations are already promoting FLOSS in many of their ICT related activities, and efforts to coordinate some of these are under way.
- *National Systems of Innovation:* There is compelling evidence that the countries which are adapting best to the emerging knowledge economy are those with the most effective systems of innovation, involving close collaboration between the public sector, the tertiary sector and the private sector. Should we not investigate, adapt for the South African context, and develop appropriate FLOSS-based innovation for development models?

The rationale behind these and other recommendations, questions and issues are discussed in the document which attempts to provide strategic and tactical pointers.

# TABLE OF CONTENTS


I INTRODUCTION 7

II "OPEN" CONCEPTS 9

WHAT IS FREE/LIBRE OPEN SOURCE SOFTWARE? 9
WHAT ARE OPEN STANDARDS? 9
WHAT IS OPEN CONTENT? 11

III HISTORY 13

IV CURRENT POLICY & STRATEGY 15

V BUSINESS MODELS FOR FLOSS 17

DEVELOPMENT 17
DISTRIBUTION 18
SERVICES 19

VI ASSESSING THE COSTS AND BENEFITS OF OPEN SOURCE 20

VII INTELLECTUAL PROPERTY RIGHTS 21

IPR AND THE ICT SECTOR 22
"COMMONS PROPERTY" - FLOSS AS A PROMINENT EXAMPLE OF A WIDENING GLOBAL DEBATE: 24

VIII APPLYING OPEN SOURCE ACROSS SOCIETY 26

A. THE PUBLIC SECTOR 26
B. THE ROLE OF FLOSS IN EDUCATION AND INNOVATION SYSTEMS 29
C. THE PRIVATE SECTOR 34

IX THE ROAD AHEAD 40

A. WHAT HAS BEEN ACHIEVED? 40
TABLE 1: Previous Recommendations Regarding FLOSS 40
B. MOBILISATION AND ENGAGEMENT 42
C. NEXT STEPS – BEYOND THE PUBLIC SECTOR 42
TABLE 2: Further Recommendations Regarding FLOSS Utilisation 46

X CONCLUSIONS 49

APPENDIX ONE: FLOSS DEFINITIONS 50

OPEN SOURCE 50
FREE SOFTWARE 50

APPENDIX TWO: LETTER FROM JUAN ALBERTO GONZÁLEZ, G.M. MICROSOFT PERU 51

APPENDIX THREE: REPLY TO MR. GONZÁLEZ LETTER FROM CONGRESSMAN NUÑEZ 53

APPENDIX FOUR: METHODS OF COST AND BENEFIT ANALYSIS 59

REFERENCES AND RESOURCES 60

## Boxes, Figures and Tables (in Section VIII)

BOX 1: Examples of FLOSS Applications: FLOSS in Public Sector Communications Technologies
BOX 2: Examples of FLOSS Applications: GCIS & Apache
BOX 3: Examples of FLOSS Applications: Language translation
BOX 4: Examples of FLOSS Applications: Rural Education
BOX 5: FLOSS Parables: Part I - Funeka's awakening
BOX 6: FLOSS Parables: Part II - Sipho's choice
BOX 7: Examples of FLOSS Applications: LegalWise Internet Server
BOX 8: Examples of FLOSS Applications: Lewis Stores
BOX 9: Examples of FLOSS Applications: Nando's and Linux
BOX 10: Examples of FLOSS Applications: Document Management in the Medical Research Council
BOX 11: Examples of FLOSS Applications: FotoFirst Data Management
BOX 12: Examples of FLOSS Applications: SME Opportunities - Kgatelopele Technologies
TABLE 1: Previous Recommendations Regarding FLOSS Utilisation
TABLE 2: Further Recommendations Regarding FLOSS Utilisation

## Abbreviations

| | |
|---|---|
| BSD | Berkeley Software Distribution |
| DFID | Department for International Development (U.K.) |
| DNA | Deoxyribonucleic Acid |
| EFA | Education For All |
| FLOSS | Free/Libre and Open Source Software |
| FOSS | Free and Open Source Software |
| FUD | Fear, Uncertainty and Doubt |
| GCIS | Government Communications and Information Services |
| GITO | Government Information Technology Officers |
| GPL | (GNU) General Public License [http://www.gnu.org/copyleft/gpl.html] |
| GNU | GNU is Not Unix – refers to the free operating system (Linux is the kernel) |
| HEIs | Higher Education Institutions |
| HTML | Hyper Text Mark-up Language |
| ICT | Information and Communication Technology |
| IP | Intellectual Property |
| IPR | Intellectual Property Rights |
| ISETT SETA | Information Services, Electronics and Telecommunications Technologies Sector Education Training Authority |
| ISO | International Standards Organisation |
| LINC | Linux Internet Network Connection |
| MRC | Medical Research Council |
| NGO | Non-governmental Organisation |
| OSS | Open Source Software |
| OSSMSP | Open Source School Management System Project |
| PDF | Portable Document Format |
| PS | Proprietary Software |
| R&D | Research and Development |
| ROI | Return on Investment |
| SETA | Sectoral Education Training Authority |
| SITA | State Information Technology Agency |
| SMEs | Small and Medium Enterprises |
| SMMEs | Small, Micro and Medium Enterprises |
| SWOT | Strengths, Weakness, Opportunities and Threats |
| TCO | Total Cost of Ownership |
| XML | Extensible Markup Language [http://www.w3c.org/XML/] |
| USA | United States of America |

# I Introduction

Information and Communication Technology (ICT) is making rapid inroads into virtually all aspects of life, affecting the way we work, live and play. Computer chips are becoming ubiquitous, spreading to industrial machinery, cell-phones, cars, household appliances and even clothing. Software contains the instructions that the chip processes in order to give these products their intended operations. Software also enables a device to exchange information with other devices connected to it via the Internet or other network. Software is thus key to the generic function of ICT, i.e. the sharing and processing of information.

Unlike the physical device to which it may lend functionality, software is a coded embodiment of several aspects of human intelligence (including information, algorithms, ideas and formal knowledge). In contrast to physical artefacts, sharing does not deplete or reduce intelligence or knowledge, but instead allows others to build upon them and generate positive learning feedback cycles. Beyond software, the sharing of human intelligence has always formed a crucial component in the advancement of knowledge. This principle has also underpinned the development of FLOSS[4] over the past 60 years.

Intimately related to FLOSS is the issue of open standards for ICTs. Increasingly, ICTs are converging in their technological functions and markets[5]. By definition, FLOSS allows access to and modification of a program's human readable instructions, i.e. the source code. As a result, FLOSS has played a major role in facilitating the development and adoption of open standards in ICTs. The development of the Internet, and its associated technologies such as the World Wide Web and e-mail, are all based on a range of open standards which evolved (and are continuing to evolve) as the industry benefits of compatibility and interoperability are more widely appreciated.

There is a commonly held misunderstanding that open access to the source code equates to open access to the applications run by the software, i.e. inadequate security capabilities. On the contrary, transparency enhances FLOSS security because peer review roots out "backdoors" and bugs from the code. Particularly in a globalising world, FLOSS and open standards allow governments and enterprises to develop and retain a level of technological independence otherwise surrendered to the owners of proprietary standards.

These advantages of openness do not apply in all circumstances. For example, an individual or organisation may have good reason not to disclose freely software that is commercially strategic or sensitive. One cannot decree that all software must be open, any more than one might insist that all knowledge must be shared freely. However, it is

[4] Free/Libre and Open Source Software. Throughout this document we use this acronym for inclusivity, to highlight the ethical underpinnings of the broader social phenomenon, and emphasise the emergence of a third mode of production in the digital networked environment: "commons-based peer-production" – beyond property and contract-based production [http://www.benkler.org/CoasesPenguin.html].

[5] The current Convergence Draft Bill addresses issues around this phenomenon from a regulatory position in South Africa. It is available at: http://www.gov.za/bills/index.html#drafts

necessary to take issue with attempts to restrict non-proprietary software use or development (e.g. through broad software patents) just as vigorously as it is necessary to defend freedom of expression and the open exchange of ideas.

Openness is also an important means by which to facilitate South Africa's active participation in the ICT sector globally. Downloading proprietary "blueprints" that inflexibly specify exact product design and assembly parameters differs considerably from an open process where alternative designs are available and compete across their performance characteristics. By expanding the scope for local innovation, an open source development environment allows local enterprises both to germinate, and to move up the international ICT knowledge value chain.

The discussions that follow begin with a review of the concepts and history of FLOSS in Sections II and III. Section IV provides an overview of current policies and strategies in FLOSS. Development models and business models associated with FLOSS in the software industry are discussed in Section V. Issues around the most appropriate metrics for assessing the short and long term benefits of open source compared with proprietary software are presented in Section VI. Questions around the balance between openness and incentives to innovate are examined under intellectual property issues in Section VII. The challenges and opportunities associated with the use of FLOSS in the public and private sectors, education and innovation systems are analysed in Section VIII. Examples of relevant success stories are also provided in this section. Section IX reviews achievements in the past two years and concludes with brief pointers to what the next steps might be on the way to realising the full potential that FLOSS can offer South African and African society.

# II "Open" Concepts

### *What is Free/Libre Open Source Software?*

Software carries the instructions that tell a computer how to operate. The human authored and human readable form of those instructions is called *source code*. Before the computer can actually execute the instructions, the source code must be translated into a machine readable (binary) format, called the *object code*. All distributed software includes the object code, but FLOSS makes the source code available as well.

Proprietary software owners license their copyrighted object code to a user, which allows the user to run the program. FLOSS programs, however, license both the object and the source code, permitting the user to run, modify and possibly redistribute the programs. With access to the source code people have the freedom to run the program for any purpose, redistribute, probe, adapt, learn from, customise the software to suit their needs, and release improvements to the public for the good of the community. Hence, some FLOSS is also known as free software, where "free" refers, first and foremost, to the above freedoms rather than the monetary sense of the word[6].

All FLOSS developers claim copyright, but then use licenses innovatively to give users a variety of freedoms. Dominant examples of these licenses are copylefted and non-copylefted (Lessig, 2003). Under the copylefted license, subsequent FLOSS developers must adopt the same license which ensures that their modifications to the code will remain open. With the non-copylefted license, developers can choose any license to cover their subsequent modifications, even a proprietary license.

### *What are Open Standards?*

Open standards in ICTs allow freedom to access as well as to contribute to the development of the standard by any interested party, which is not possible under a proprietary standard. Standards are *de facto* and *de jure,* and both arise out of complex dynamics influenced by economic, political and social forces. However, a *de facto* standard emerges informally as a significant number of players find it useful, whereas a *de jure* standard is created via a formal process involving authoritative bodies and/or legislation. Indeed *de facto* standards can be, and have been formalised as *de jure* standards, but this is not always the case. Three examples are provided to help clarify this important issue:

- TCP/IP (Transmission Control Protocol/Internet Protocol) is the crucial suite of de *facto* standards upon which the internet was developed. It evolved over several years (first published in 1981) via the open process of the exchange of snippets of source code between many interested parties. The equivalent *de jure* standard is the ISO/OSI network model. X.25 is an example of the implementation of this model. Twenty years ago, although the Internet was

[6] See http://www.gnu.org/ and APPENDIX ONE: FLOSS Definitions for further discussion.

emerging all around them, the IT industry and most corporates were investing in OSI, not TCP/IP[7].

- OASIS is a consortium developing *de jure* open document standards, including XML-based standards for open office formats, biometrics, e-business, and e-government[8]. Prominent proprietary alternatives (e.g. Microsoft Office and WordPerfect) do not comply with these open standards, causing incompatibility between each other and the FLOSS applications which do comply – thereby inhibiting end users around the world and undermining the spirit of open standards for universal interoperability.
- HTML (Hypertext Mark-up Language) is another high impact *de facto* open standard. Essentially, HTML pages are open source code. It is widely accepted that the phenomenal growth of the Web in the middle of the last decade would not have occurred had this code not been freely available for any Web 'surfer' to copy, experiment with and use.

Open standards in ICTs are critical to allow new entrants to participate, innovate on standards implementation, and compete. With a proprietary standard, the owner can prevent competitors and entrants from capturing market share through their legally enforceable IPR (see section VII). In a developing economy like South Africa's proprietary ICT standards are typically held by foreign enterprises which effectively relegate domestic engagement to the level of franchisee. In contrast, a genuinely open standard that is true to the FLOSS philosophy facilitates alternative, interoperable, programs. This sort of open standard decreases public sector dependence on particular vendors. Services, support and maintenance can be sourced to a range of suppliers with this type of open standard, which increases market competition and opportunities for local involvement in the value chain. Additionally, a variety of alternative vendors increases the scope for future strategic decisions.

There are examples where an organisation or consortium allows access to their software's source code, but severely limits permission to modify or adapt that source code. These restrictions effectively prohibit the development of compatible alternatives or open standards, and should not be confused with FLOSS and open standards initiatives (e.g. see Microsoft's shared source initiative[9]).

It cannot be over-emphasised that the Internet and World Wide Web, which are having so much impact on the world today, would not exist without FLOSS and the inter-related development and adoption of the essential open standards. For example, the most basic Internet applications such as e-mail, FTP, Gopher and the World Wide Web all make use of FLOSS for their successful deployment and overall adoption. Today,

[7] With hindsight this appears foolish, yet we need to understand the factors influencing the conventional thinking of a large number of intelligent people at that time. Perhaps history is repeating itself today, with important, innovative opportunities (including FLOSS) being inappropriately marginalised and delayed.
[8] See "http://oasis-open.org".
[9] See "http://www.microsoft.com/resources/sharedsource/default.mspx"

nearly two-thirds of all web servers use FLOSS[10]. For the growth in services, inclusivity and innovation to continue anywhere in the world, it becomes all the more important to maintain the established open approach to the further development of the Internet and World Wide Web.

### *What is Open Content?*

Currently, open content[11] is treated as the poor relation in the 'open' family (e.g. the government's strategy on FLOSS does not directly address the issue of open content). However, it is an essential and equal partner to FLOSS and open standards, and, although it is only briefly discussed here, merits much wider understanding and discussion. In this context, "content" stands for all and any types of creative human made material (whether text, pictures, audio, video, etc., and usually digitised). That content is labelled as 'open' if it is made available, or published in ways that allow it to be re-used, copied or redistributed without charge (beyond handling) via a non-restrictive format and copyright license[12].

Open content is believed to have many benefits, the most obvious being the lower overall costs as a result of the avoidance of duplication of content creation, management of content, collaboration, quality via peer review, and improved access to information. However, possibly more important benefits include:

- the promotion of more transparent, efficient and effective governance, where it has been adopted in the public sector, and
- the lowering of barriers which limit the production and use of language and culturally sensitive local content – of particular importance in the developing world.

Although the migration of an organisation to open content standards is unlikely to incur significant costs or technical complexity, it is likely to have a major impact on the mindsets of both the user community and the ICT service providers, and subsequently on related organisational and business processes. The South African government is already moving towards open content by adopting MIOS ("Handbook on Minimum Information Interoperability Standards)", with the aim of allowing information to flow seamlessly across government systems. However, to date there are very few initiatives where these standards are being implemented or enforced.

Today, there appears to be a transformation of the world economy from industrial to knowledge-based.[13] Granstrand (2000) refers to this economic transformation as the emergence of intellectual capitalism, where ICTs are playing a pivotal role by lowering the transaction costs associated with the exchange of information, enabling a more

---

[10] Netcraft's survey http://www.netcraft.com/survey shows that some 65% of web sites use the open source Apache Web server

[11] See Wikipedia, http://en.wikipedia.org/wiki/Open_content

[12] Several Open Content licenses exist, e.g. the GNU free Documentation License (GFDL) is a copyleft license for free content, designed by the Free Software Foundation (FSF) for the GNU project. The official version 1.2 of the license can be found at http://www.gnu.org/copyleft/fdl.html. The Creative Commons approach permits generation of custom licenses according to which rights the creator wishes to retain/ give away: http://creativecommons.org.

[13] For an overview, see the special volume in the *International Social Sciences Journal* (v171, 2002).

adequate privatisation of gains from production, and making possible the distribution of information on a commercial basis. ICTs augment our ability to codify information[14], as well as enabling us to present it in a wide variety of forms, and to distribute it widely. However, ICTs also provide a variety of mechanisms that enable us to limit and exclude access to information. The software and standards that drive these ICTs into the future can shape the emergent knowledge economy, and particularly influence who is included or excluded. Hence, FLOSS, open standards and open content are issues whose importance goes far beyond the boundaries of the software industry alone[15].

[14] For a study on the role played by ICTs in codifying knowledge see Cowan (2001).
[15] See: http://www.benkler.org/CoasesPenguin.html for a deeper exploration.

## III History

Free or non-proprietary software has existed since the invention of the first computers in the mid-1940s, and for many years it was the norm. However, its production, distribution and use were limited to a few engineers, scientists and others who had access to the then expensive and scarce computing facilities. In the universities and public sector (particularly military institutions) where these computing facilities were available, the free exchange of software was facilitated by the programmers being paid for the act of programming and not for the programs themselves.

While FLOSS includes many types of programs, its history is tied to the emergence of a portable computer operating system. Until the early-1970s, operating systems were written for specific mainframe and mini-computers which were not intended to be compatible, and hence software had to be modified for each family of machines. Thus, proprietary software used by IBM, Burroughs, Honeywell, and other large computer makers helped to differentiate them and establish areas of brand dominance.

In the early 1970s, programmers at AT&T Bell labs developed the UNIX operating system using the C programming language, which could be ported to any proprietary machine. Rapid adoption and diffusion of UNIX resulted from its performance and portability, as well as AT&T's decision to license it on very favourable terms to all universities in the USA.

In the early 1980s, the computer revolution took off with the spread of the personal computer and the exponential expansion of its use in a growing range of business processes, as well as computer chips being embedded in more and more products. As computers expanded into many business sectors, programmers were no longer paid to program as in a research environment, but were paid for the programs themselves. This applications focus quickly expanded the amount of proprietary relative to non-proprietary software. Around the same time, AT&T began to increase the licensing fees for its UNIX operating system, leading in 1984 to an initiative by Richard Stallman at MIT's Artificial Intelligence Laboratory against the proprietisation of software. Stallman's pioneering efforts with the GNU project and the Free Software Foundation have generated several successes. In particular, they paved the way for perhaps the most notable 'open' success story to date, i.e. the development of GNU/Linux[16] - an open UNIX-like alternative[17] – which was released in 1994 after Linus Torvalds' contribution of an operating system kernel to the GNU project resulted in a complete system.

In the 1990s, the spread of the Internet stimulated the development of free software communities around the world, interested not only in operating systems, but also in many applications. However, while non-proprietary software showed highly competitive performance, there was limited commercial backing. Attributing the commercial

[16] Strictly speaking, Linux refers only to the kernel of the operating system. Linux is often used to refer to both the kernel and the GNU/Linux operating system.
[17] See Moody (2001) for a history of GNU/Linux.

scepticism to the "free" label, in 1998, the free software community split[18] into the Free Software Foundation and *open source*[19]*.* Today, FLOSS is a rapidly growing and commercially accepted alternative to proprietary software across a range of systems and applications. However, the free software community remains an important movement, separate from the FLOSS community, and emphasising a range of ethical issues[20].

It is interesting to recognise the crucial role the government of the USA has played in stimulating the development of open standards and freely available software for decades via funding agencies, oversight of standards and procurement regulations. Although few appreciated it at the time, these actions were creating the foundations for both the global ICT industry, and the phenomenon of globalisation. Yet in the '80s and particularly the '90s, support for FLOSS in the USA was dramatically reduced. In contrast, outside the USA, many other governments are realising the global and national importance of open standards and FLOSS, and their numbers and levels of support are growing annually.

Recently, the Free Software and Open Source Foundation for Africa (FOSSFA) and a number of partner organisations convened a conference hosted by the University of the Western Cape: "Idlelo, the first African conference on the digital commons". This meeting was content rich and brought together a diverse range of people working in the areas of FLOSS and Open Content. A number of follow up events took place. One of these was the Africa Source meeting in which some of the continent's best ICT brains got together to exchange their knowledge in FLOSS. Among other things, this meeting saw the release of a Kiswahili spellchecker. In South Africa, a number of innovations have been taking place. These include the launch of the first localised GNU/Linux distribution.

In 2004 FLOSS has received unprecedented attention in Africa, with governments developing policies, donor agencies taking note, and industry responding with both support and scepticism.

[18] See http://www.bitmover.com/poster/poster90.gif
[19] See Raymond (2003 and 2001) for a history of this transformation of the free software community.
[20] See http://www.gnu.org/philosophy/philosophy.html

## IV Current Policy & Strategy

There is a groundswell the world over of adoption of FLOSS by individuals, as well as a wide range of organisations, whether small or large, public or private. This reflects a growing acceptance of and confidence in FLOSS and open standards, which, in turn, is contributing to their further and faster development. It is no idle matter that leading players in the ICT industry such as IBM, HP, Sun and Oracle have expressed their confidence in, and commitment to FLOSS[21].

Indeed, South Africa is the first African nation to follow the example of many nations in developing policies to promote the use of FLOSS. Examples include China, Japan, South Korea, Thailand, India, Israel, Argentina, Brazil, Peru, Germany, France and the United Kingdom[22]. ICT professionals are also encouraging their public sectors to adopt policies that support FLOSS, New Zealand being a notable case in point[23]. Many of these countries are recommending or legislating that FLOSS should be used by the public sector and its associated agencies, unless proprietary software is the only available option.

The initiatives in Germany, France and the UK are closely allied to the European Commission's initiative "eEurope – An Information Society for all". The initiative's action plan sets the target*:*

> *"During 2001 the European Commission and Member States will promote the use of open source software in the public sector and e-government best practice through exchange of experiences across the Union"*.

In 2000, the Commission published a document on free (libre) software analysing the phenomenon and making recommendations on *"how to help Europe to benefit from open source software"*[24]*.* A FLOSS report was completed for the European Commission in June 2002[25]. The report emphasised the importance and role of FLOSS in the economy and assessed its impact on policies and decision-making. It developed indicators of value creation within the FLOSS environment, and highlighted dependencies on the FLOSS development community. The study also identified and appraised business models and best practices in the FLOSS community.

The promotion of FLOSS by the public sector in any country is likely to cause contention which needs to be openly, widely and thoroughly debated. A good illustration of this is a recent initiative in Peru. The General Manager of Microsoft Peru produced a detailed criticism of that nation's legislation to promote FLOSS. In reply, Peruvian Congressman Nuñez argued that the Peruvian constitution guarantees public access to information that is only possible through FLOSS. As the Peruvian debate is illustrative of many of

---

[21] "We believe very, very strongly that open standards and open-source software are absolutely critical foundations for the IT business going into the future." IBM, http://www.ibm.com/news/us/2001/08/15.html

[22] See http://www.openia.com/opensource/governments

[23] See http://www.openz.org/

[24] "Free Software/Open Source: Information Society Opportunities for Europe?", http://eu.conecta.it/

[25] See "Free/Libre and Open Source Software: Survey and Study", http://FLOSS.infonomics.nl/ as well as http://flossproject.org/flosspols/

the issues involved in adopting FLOSS (several of which are of relevance beyond the public sector), translations of these letters are included in Appendices Two and Three.

# V Business Models for FLOSS

When considering the challenges and opportunities of FLOSS, it is useful to discuss business models in terms of three basic components of the software industry: development, distribution, and services, as well as noting the two broad categories of software:

1. Infrastructure: the 'plumbing' of ICT systems and the Internet, which includes operating systems, databases, Web servers and other components that enable software applications to run. This category includes Web and system middleware, the increasingly important intermediate layer between the lower levels of infrastructure and user level applications.
2. Applications: including generic desktop applications such as browsers, word processors, spreadsheets, financial and management systems, as well as various other (often highly specialised) applications that an organisation may need to run its business.

***Development***

FLOSS development is very distinct from that of proprietary software. The comprehensive FLOSS study (2003) provided evidence that FLOSS programs are often produced by core groups of volunteers from all corners of the globe[26]. This is possible because the accessibility of the source code allows efficient scrutiny of the latest versions which can be distributed widely via the Internet. In this process, FLOSS development in many ways parallels the scientific method of peer review, and has resulted in robust software, well known for its reliability.

Raymond (2001) refers to this distributed volunteer FLOSS production model as a "Bazaar", in comparison with the "Cathedral" of the orthodox hierarchical development model[27]. An alternate characterisation is the "cooking pot" model, where single ingredients merge to create a product of significantly more value than the sum of the individual ingredients (Ghosh, 1998).

Of course, along with its strengths the volunteer driven FLOSS development model has also exhibited weaknesses. Recent analyses have revealed that the reward structure of volunteerism is limited in its ability to provide incentives for wearisome tasks like documentation and on-line support[28]. Yet these are critical components from a software user's perspective, particularly in applications. The emergence of hybrid open source business models (see below) and the increasing engagement of traditional proprietary

[26] For an excellent overview of OSS development see the special issue of the journal *Research Policy* (vol. 23, 2003). The seven articles in the special issue examine three major topics in open source development: 1) motivations 2) the innovation process and 3) competitive dynamics. Through these headings the implications of OSS development on areas ranging from the economics of innovation to organisational structures are explored.

[27] See http://www.catb.org/~esr/writings/cathedral-bazaar/

[28] See Bonaccorsi & Rossi (2003), FLOSS (2003), and Lerner & Tirole (2000). Nonetheless, Lakhani and Von Hippel (2000) hold that the low costs of communication through the Internet facilitate OSS support.

software firms in FLOSS initiatives suggest two possible means by which these weaknesses in the FLOSS development model are being addressed.

The volunteer model of software development evolved in the developed nations of the North, where wealthy institutions, well-funded universities and corporations support collaborative provision. The question is how this model translates to the economically poorer countries of the South and how it should be modified, if need be, without killing the goose that lays the golden egg. In South Africa, consensus is growing that public support is needed for the development of FLOSS to meet national needs. This suggests a hybrid development model involving distributed volunteers anywhere in the world and explicit financial support for South Africans with the enthusiasm and aptitude to be involved in chosen development projects. Well-conceived projects can be an excellent mechanism both for capacity building in FLOSS and for assimilating the discipline of collaborative software development.

FLOSS has been the *de-facto* standard of academia for many years (particularly in the USA), with significant resultant technological benefits and impacts in the national ICT industry. Based on this example, a recent analysis of FLOSS in the UK recommended that FLOSS should be the "default exploitation route for Government R&D software" (Peeling and Stachell, 2001). Many believe that this recommendation should be adopted by South Africa for software R&D funded by the public sector and conducted by its universities and research councils. Such a strategy would have several benefits, and would be in the interest of South Africa's industrial competitiveness. Additionally, public sector agencies should adopt the FLOSS development model whenever possible.

It would also make sense for the public sector to co-operate with other nations and international agencies to develop interoperable software standards and systems that are of common interest. An example is the UK's involvement in the development of XML schemas (open, web-based information exchange standards) as part of an e-government interoperability framework (Cabinet Office, 2001). Nations could also share experiences on various projects, such as FLOSS desktop deployment pilots.

An essential complement to the FLOSS development model is evaluation and certification. The proposed hybrid development model might include a co-ordinating agency with longer-term employees to carry out this function on behalf of bodies such as the State Information Technology Agency (SITA). It is expected that the training and support role would be partially satisfied by a groundswell of companies stimulated by government's commitment to FLOSS.

### *Distribution*

The co-operative nature of the FLOSS development models described above limits the value-addition that can be internalised by individual organisations at the development stage. Therefore, if an enterprise wants FLOSS to form its core business it must look downstream towards FLOSS distribution and/or FLOSS-related services[29]. In

[29] This discussion draws substantially from the analysis of OSS-based business models in the FLOSS (2003) reports.

distribution, we can make a distinction between distributors of the various suites of products centred on the Linux operating system kernel[30] and distributors of other FLOSS products which have yet to match the global penetration of GNU/Linux[31]. Among GNU/Linux-based distributors, packaging is an area of added value with low margins per unit sold. Branding and exclusive sales channels bolster these revenues, but increasingly distributors are also engaging in FLOSS-related services where they face enterprises that often use combinations of open and proprietary software. In contrast, distributors of other FLOSS products are typically in niche markets and are dependent on value added resellers and/or original equipment manufacturers. Hence, to date, pure FLOSS distributors in products other than GNU/Linux are rare. The non-GNU/Linux FLOSS distributors that do exist often engage in additional consulting and support services.

***Services***

There is growing demand for a variety of FLOSS-related services, ranging from systems integration to support and training. Again, enterprises that are primarily from a FLOSS background must compete against others whose capacities are more generally in ICT services and who often use combinations of open and proprietary software. The FLOSS (2003) study of FLOSS-based business models uses this division as a primary categorisation of firms active in FLOSS-related services. Among the firms with FLOSS backgrounds, their competitiveness often resides in substantial product and technology knowledge, which bolsters their services business. Firms solely based on FLOSS products exhibit the highest dependence on the acceptance and development of FLOSS. The success of firms in this general group depends upon the relative importance of product know-how in the services demanded. Product support and training are examples of areas where product know-how is relatively important and in which process know-how can be acquired easily.

Firms without FLOSS backgrounds depend upon that very process know-how in the ICT services sector to compete in FLOSS-related services. Substantial process know-how is particularly important in strategic consulting services where high level product know-how can be acquired with relative ease. The FLOSS (2003) analysis concludes that firms' success will depend on the market for and their positioning in FLOSS-related services that match their relative strengths, i.e.:

- *process know-how* in the case of firms without strong FLOSS backgrounds, and
- *product know-how* in the case of firms with deeper FLOSS backgrounds.

In South Africa (as in much of the world) it is clear that the market for FLOSS-related services is in its early stages of development. How best do we stimulate that market?

[30] All Linux-based distributions include a version of a common Linux operating system kernel since there has been no forking. New versions of Linux are developed via a system of series numbering, odd numbers being the pre-stable stage, where innovations and improvements from many sources are tested, even numbers being restricted to stages once stability of these additions has been well established throughout the global OSS community. Those concerned that open processes cannot be efficient and effective should study the Linux versioning mechanism in some detail.

[31] Examples include Mozilla, Zope, Apache web server – see http://theopencd.org

# VI Assessing the Costs and Benefits of Open Source

Probably the most immediate attraction of FLOSS to most people is that its use does not incur the high licensing costs associated with proprietary software. However, as with proprietary software, there are costs involved in implementing FLOSS solutions associated with the time and skills of local ICT technicians. Importantly, the funds covering these latter costs flow into and circulate within the local economy, whereas most of the licensing fees leave the country.

Many insist that FLOSS must prove itself to be a viable alternative to proprietary software. The question most commonly posed is whether proprietary license costs are truly greater than the lifetime implementation, maintenance and support costs associated with FLOSS. Significant resources have been invested in answering this question with a battery of objective, fact-based, cost/benefit tools (see Appendix 4). SITA has recently provided a detailed guide for undertaking such financially based analyses to assess potential implementations of FLOSS[32].

The well established FLOSS products generally do well in such analyses. However, this limited focus on financial metrics is potentially dangerous since the wider social implications tend to be ignored, even though this longer term arena is where the greatest benefits of FLOSS mostly lie. For technology in general it is true that most attention is focused on their economic costs and benefits in the short term (the proximate effects), whereas the full costs and benefits can only be assessed across economic, social and environmental dimensions over the short, medium and long terms (the ultimate effects). This is particularly the case for FLOSS in South Africa where, in addition to the consumption of existing FLOSS products, emphasis is growing on the local development (and local involvement in global development) of new FLOSS products.

Given the broader, national, cultural and long term advantages, such as reducing the country's dependence on imported software, and opportunities for local innovation, the FLOSS route is the logical way to go. The challenge is to level the playing fields by raising awareness of this issue and encourage those who support and strive for the medium to long term goals.

[32] See Blume et al (2003).

## VII Intellectual Property Rights

Software is but one of the areas on which IPR have significant impact. However, recognition is growing that better understanding by a broader range of stakeholders is needed of the complex relationships between IPR, the information society as a whole, and the development, distribution and services aspects of the ICT industry.

One of the first forms of IPR were patents, introduced in Venice in the late 15th century and then in England in the 17th century (David, 1993). Today, there are four principal IPR mechanisms: patents, copyright, trade secrets and trademarks. Throughout the past century there has been an increasing relative importance of intangible capital and innovation in the highly competitive global economy (David and Foray, 2001). It is important to emphasise, therefore, that today's principal IPR mechanisms are not well adjusted to intangible capital and the emerging knowledge economy, since they were conceived in an era when technological advances embodied in physical goods were the norm.

These days, IPR mechanisms stimulate significant disagreements among a variety of stakeholders, with some camps advocating a relaxation of some components of IPR while others advocate their strengthening. Most debates assume a fundamental and universal trade-off between short-term knowledge creation and medium-term knowledge diffusion (Gilsing and Van der Steen, 2003). Within and across sectors, a division exists between knowledge produced for private gains and that produced as a public good.

- *Private Gains:* Where the quest for private gains predominates, IPR are crucial to investment in and production of new knowledge[33]. In this situation, IPR encourage continued knowledge creation by granting a monopoly over the exploitation of the resultant intellectual property. However, IPR are typically for a set duration, after which the intellectual property becomes part of the public knowledge commons. The idea is that the issuance of short-term monopoly rights facilitates knowledge diffusion in the medium-term by decreasing the tendency to retain knowledge as trade secrets. However, estimating the costs versus benefits to society (both direct and indirect, as well as short and long term) is proving to be a far from straightforward policy exercise.
- *Public Goods:* When the goal is the generation of public knowledge, the reward structure encourages the production of intellectual property as public goods without depending on the incentives of IPR[34]. Incentives in this case include a combination of priority and reputation rewards that may create disclosure races.

It should be noted that prominent figures in the FLOSS communities highlight passion, fun and creative expression in the art of software development as primary motivators (e.g. Torvalds and Diamond, 2001; Himmanen *et al* 2001). The stance of the Free Software Foundation [http://www.fsf.org/philosophy/] provides a possible exit point for

[33] In addition to IPRs, software developers can "protect" their IP through release of the object code only and through copy protection technologies.
[34] See Stephan (1996) for an excellent overview of the public knowledge production system.

the debate by questioning the foundation of what we have referred to as IPR. Policies which have an impact on socio-economic development must consider this debate carefully, and in the current South African context, err on the side of "freedom". The challenge is to facilitate meaningful participation by all in the debate to raise awareness and understanding of such issues.

### *IPR and the ICT Sector*

As the impact of FLOSS grows, it is not surprising that IPR are becoming increasingly contentious in the software industry since proprietary software development tends to depend upon private gains, while FLOSS development tends to utilise the incentives found in the production of public goods. How far are we from the ideal balance? There is growing belief that in software and information goods in general, increasing knowledge diffusion, at the possible cost of incentives to innovate, increases the net social value (Cowan and Harrison, 2001a). Decreasing the strength of the IPR system, therefore, should advance social benefits. ICTs, and most notably software, are cross-cutting 'general use' technologies, contributing in a variety of ways to many other technologies or goods. Hence, if software were widely diffused it should have very large positive effects on further knowledge creation as well as lowering the costs of production of many goods in the economy. In addition, Cowan and Harrison (2001a) point out that the cost of strong IPR (in the form of monopolies) has also increased by preventing entry, which in the rapidly evolving ICT sector is vital for continuing innovation. Strong IPR thus prevent the type of virtuous entry that causes and perpetuates technological advancement in the sector.

*Are we on the right track in terms of freedoms for innovation in software development?*

A publication by the National Research Council (2000) in the United States illustrated the 'digital dilemma' through the example of patenting of software, something that is not (yet) possible in South Africa. Rather than encouraging software developers to make their work public, IPR appear to have created the opposite effect. The patenting of software is questionable for several reasons, including:

- Software patents correspond to the patenting of ideas and algorithms (which are general and universal) rather than inventions in the traditional sense.
- Software patents are granted for extremely long periods relative to the ICT innovation cycles, which are typically of the order of 18 months.
- Software patents have frequently been granted for simple notions, leading to a plethora of notorious 'junk patents'.

Consequently, it is currently very difficult to write any significant piece of proprietary software that does not infringe on a number of questionable patents granted in the US[35]. This effectively enables a small cartel of large corporations (mostly in the USA) to prevent access to the market by new entrants. Large company A (which owns a large stock of patents), generally comes to an amicable agreement with large company B (which also has a similarly large stock of patents) in the development of a new software product. However, a new entrant or start-up (perhaps from developing country C) has

[35] Recently, Microsoft has been involved in a battle to invalidate just such a patent on a program that lets a Web browser summon programs automatically over the Internet (Lohr, 2003).

no such stock of patents with which to bargain, and therefore must deal with every infringed patent (usually incurring significant expenditure of time and money) before its software product can be placed on the market[36].

In our modern era of globalisation, efforts to harmonize IPR mechanisms, including software patents, have contributed to an international creep in their strength. This occurs because nations tend to resist the release of their IPR in harmonization negotiations, leading to a cumulative growth in IPR strength[37]. The implication is that intervention is required for South Africa.

*How far are we from the ideal freedoms in software distribution?*
Globally, the market for workstation operating systems is dominated by Microsoft's proprietary Windows products. As a result, in South Africa it is extremely difficult to obtain a new PC without Windows being pre-installed, irrespective of the wishes of the user. This is the case whether PCs are obtained from dealers, retailer outlets, or within corporate environments. At present there is no adequate regulatory framework to deal with this or other software issues in South Africa (and many other countries). However, it seems clear that for FLOSS to be given an equal opportunity to establish itself, regulation is required until all PCs can be purchased with the operating system and applications of the users' choice loaded, rather than the vendors' preference.

*How far are we from the ideal freedoms in ICT and related information services?*
Software is a central component of ICTs, which in-turn affect knowledge creation throughout society by facilitating access to large quantities of information and by facilitating interactions within and among technologies, developers and users (David and Foray, 2001). Although ICTs have the potential to improve access to information enormously, they also provide opportunities to inhibit access to information which were not previously practical (David, 2000). Excessively strong IPR risk is draining the vitality of the global public science system (NRC, 2003)[38]. Therefore, a new tragedy of the commons, called the 'anti-commons', is thought to be developing because of the over-fencing of the knowledge commons to extract economic rents (Heller and Eisenberg, 1998).

The area of content and IPR is closely linked to the principles of FLOSS (see section II). Of all aspects of the sector, information services probably impact a far greater range of stakeholders in a more tangible manner, including all facets of government as well as other public and private sector organisations. There are several IPR mechanisms associated with content and information services, including authorship, publication, modification, use, buying and selling. However, Van der Merwe[39] points out that the digital phenomenon "makes a nonsense of the different categories of intellectual

[36] For a discussion of the impact these patent clubs are having see Grossman (2002).
[37] In September 2003, the European Parliament approved legislation that permitted the granting of patents for computer-related inventions. While opposed because of its seeming expansion of IPR, it was supported as a necessary step to bring an increasing degree of harmonisation between the European and U.S. patenting systems.
[38] Also see NRC (1999 and 1997) for further discussion of issues around the access to data and information.
[39] For an interesting South African view on this complex issue, see van der Merwe (1999)

property" still adhered to in copyright legislation world-wide, "since in digital form the different works may be seamlessly integrated with each other to form a so-called 'multi-media' product, which combines text, pictures, sound and video". He also warns that "traditionally, the conservatism of the law has been beneficial and has led to certainty and predictable results in the courtroom. In the digital era, with its exponentially increasing speed of development, it means that the slow-moving law may never catch up with the matters it is supposed to regulate." This is particularly the case in South and Southern Africa, where the current copyright laws are outdated, and the legislation needs upgrading to take into account the potential benefits of new electronic media[40].

***"Commons property" - FLOSS as a prominent example of a widening global debate:***

The above sections focus primarily on the strengths and weaknesses of "proprietary" IPR, with FLOSS implicitly filling the remaining "non-proprietary" grey area. The danger of thinking of FLOSS (and other commons property – or public goods) simply in terms of being "non-proprietary" is not only that it is as inappropriate as describing women as non-men, but it also leads to the widely held misunderstanding that FLOSS is a single, coherent entity. In fact, it is as complex and rich an area as the proprietary IPR domain – for example:

- The long running debates around GPL style licenses compared with BSD style licenses compared with the public domain.
- OSS has often been incorrectly used by default to refer to both OSS and Free Software. Instead the arguments for Free Software should be considered separately since they go beyond the practical bottom-line considerations and add the necessary dimension of good governance, business ethics and other important philosophical considerations.

One of the more exciting aspects of the new thinking around commons property (taking FLOSS as inspiration) is the conscious attempt to define it as something of value in its own right (James Boyle, 2002). Until quite recently, the protection of the public domain – the intangible commons – was an essential component of the approach to IPR by most nations. But now the vision appears to be that property rights should be established much more widely, e.g.:

- Expanding patentable and copyrightable subject matter.
- Lengthening the copyright term.
- Giving legal protection to "digital fences", even if it is used in part to prevent fair use.

Each of these examples appears to be based on the assumption that human production left open to free use is inefficient and ineffective. Properly applied, IPR promise an excellent decentralised system for the promotion of innovation. The concern is that enclosure of the information commons via the significant recent expansion of IPR may not simply hamper the non-proprietary mode of intellectual production, but runs the risk of ruling it out altogether. Given the crucial role innovation plays in both economic

[40] A useful guide to all legal aspects of ICTs is provided by the 2003 guide on Cyberlaw by OSISA and Cheadle, Thompson and Haysom.

development and poverty alleviation, it is recommended that patents on standards, software and algorithms be opposed. In terms of open content, the Creative Commons approach (http://creativecommons.org) should be considered, understood and adopted as appropriate. South Africa needs to understand and manage this unacceptable risk across all sectors of society.

## VIII Applying Open Source Across Society

While many people believe that the future impact of FLOSS in the ICT industry and in society in general will be profound, it is necessary to be explicit about the challenges and opportunities associated with applying FLOSS. Attention is first turned to the use of FLOSS in the public sector. Next, FLOSS uses in education and innovation systems are discussed. Finally, we review usage of FLOSS in the private sector.

### *A. The Public Sector*

The use of FLOSS in the provision of public services and information can be examined in three categories: administrative use, accessibility, and domestic capacity building. FLOSS utilisation in the public sector is also an option to help increase technology transfers from academia. Because of the inherent ability to modify FLOSS, a program developed in academia has a better chance of being incorporated into an FLOSS system in the public sector than into a proprietary software program. Similarly, FLOSS can facilitate increased technology transfers from academia to the private sector. Thus, the FLOSS approach is potentially important in facilitating technology transfers across sectors.

#### *Administrative use of FLOSS*

Uses of FLOSS in public sector administration are similar to issues around its general use as a work tool in the other sectors. FLOSS has not yet had a significant impact in the generic desktop applications category, where Microsoft Office, running on the Microsoft Windows operating system, predominates. Nevertheless, there are already many open source office suites, such as KOffice, Gnome Office and Open Office, which offer equivalent functionality and ease of use. Hence, obstacles to the adoption of FLOSS desktops no longer include technical issues or ease of use, but seem to be based on a lack of awareness.

**BOX 1: Examples of OSS Applications: OSS in Public Sector Communications Technologies**

The use of OSS in the government of South Africa dates as far back as 1994. The government was required to set up e-mail and domain name service in OpenNet, which was created under the auspices of the Central Computer Services (CCS) a division of the Department Of State Expenditure. In 1997, the expansion of the network and the growing numbers of users requiring Internet access and e-mail services necessitated the acquisition of more and faster servers which were installed with OSS systems. Since then, several other OSS upgrades and changes have been made.

Today, open source operating systems continue to be used in network infrastructure. Stability and reliability have been impeccable. For example, last year, a server in Port Elizabeth had clocked up a continuous runtime of 637 days without any application or system downtime before it was rather impolitely restarted due to a power failure. The OSS systems provide network bandwidth and connectivity to national, provincial and local government and also provide Internet and electronic mail services to the rest of its users.

OSS provides a powerful resource and has demonstrated its usefulness through the services it has provided to the South African government. The benefits that have been realised out of this effort include:

- Cost savings on licensing and upgrades
- Increased levels of security
- Improved response times in recovering from system failures

It is reasonable for government to seek full rights to order or commission development of software to their own specification rather than allow the contractor to control this area. Where appropriate, government can choose to release the software under an open source license and hence open the contractor's work to general scrutiny. Maintenance and upgrade might then be conducted by different contractors and hence minimise supplier lock-in.

The MS Office example illustrates the challenge that proprietary technology creates. When software is 'closed', it is difficult for competitors to develop compatibility. Externalities in compatibility are important forces in establishing open standards where, as the network of users grows, the value of its interoperability rises in a self-reinforcing manner. Conversely, if the software is closed it leads to public sector dependence on particular vendors and the potential for externally imposed upgrades with associated additional licensing, training, and hardware costs.

This also creates a problem in terms of innovation. Other organisations are unable to develop improved products, complementarities and add-ons when proprietary software and closed standards predominate. Particularly as software is often a complex technology, this is likely to hinder innovation in the long run (Cowan and Harrison 2001a)[41].

Administrative use of FLOSS goes deeper than the desktop and includes applications such as payroll systems and workflow including leave management, cash claims, human resources management, finances, etc.). Components of these systems might include proprietary products such as Oracle which require a large investment in time and money to set up and establish. The challenge of migrating legacy systems to open source platforms can be daunting and involve risks many companies will hesitate to take without clear guidelines and support.

### *Accessibility and FLOSS*

FLOSS is an important option to consider in the provision of access to public information. Open standards and open source provide a transparent basis upon which to store securely and maintain the integrity of public data. They also act to ensure that public access to online government documents is facilitated (see box 2 and Appendix 3).

**BOX 2: Examples of OSS Applications: GCIS & Apache**

Government Communications and Information Services (GCIS) have been running a web server since 1995. After some research, it was decided in 1998 to evaluate the popular Apache web server. The Apache source code was downloaded, compiled and installed as a test server. The installation and configuration was surprisingly easy and soon Apache became the platform to host the site. Besides better server performance, Apache was more flexible and easier to administer. An added benefit was that downloading was the only cost.

Upgrading the server over the years has always proved to be a smooth experience, not even requiring a reboot. As more functionality was required, it was also easy to implement new modules. Several other open source solutions like perl, PHP, Tomcat, Python, Mailman list server and a local mySQL database server have been implemented over the years and are all running together with Verity's search engine on a single server.

Currently, the server is hosting 24 virtual hosts (www.gov.za being one of them) with an average of 107 746 hits a day (figures for October 2003). Server uptime is more than 99.99% with the national network being the biggest bottleneck. Net-based support has been excellent, with queries from all over the globe answered in minutes.

[41] See also Reichman (1994).

The modern public sector generates vast numbers of digital files like birth certificates, tax returns and DNA records of criminals that must be retrievable in perpetuity. As these digital records are likely to continue to expand further through the promotion of e-government there is a strong incentive for the public sector to avoid lock-in to proprietary formats to store this data. Modernising and ensuring communication and interaction across divisions in the public sector, e.g. by the use of such freely available formats for document exchange as HTML and PDF, is critical to realising the vision of a coherent 'joined-up government'. Indeed, this is a fundamental recommendation of this document.

### *Capacity Building and FLOSS*

Domestic capacity building in the ICT sector is another area where the use of FLOSS can play an important role. Familiarity breeds resistance to change, even for users who are not too satisfied by the current offering (i.e. "better the devil you know…".). Newcomers to computers, particularly students and youth, are likely to be more receptive to a different, open model. Furthermore, because the source code is available, the FLOSS desktop can be customised freely to suit local needs. An obvious candidate is support for local languages. This alone makes a compelling case for the FLOSS desktop in South Africa (see box 3). Accordingly, it would make sense to pilot the FLOSS desktop in selected public sector institutions as a possible precursor to more widespread use.

The language support problem is not particularly significant in most developed countries where 'mainstream' languages dominate and are supported by the major proprietary package producers. Furthermore, in a developed country, familiarity with Microsoft Office is more entrenched than in a developing country. Against this background, there would appear to be a greater opportunity for the FLOSS desktop to establish its value in the developing world. Even so, there are informative cases of FLOSS desktops being adopted or piloted by institutions in the developed world[42].

**BOX 3: Examples of OSS Applications: Language translation**

South Africa has eleven official languages. Hence the communication of decisions and policy to the nation is a logistical nightmare. And what of the learner who's English is not very good but who is fluent in Xhosa? Most of our population is excluded from current ICTs simply because they are not fluent in English. The old solution is to improve English literacy, but the alternative of providing software environments in several African languages is now both possible and simple.

If it is so easy then why are there no African language software tutorials, online manuals and associated software? The main reason is that there is little commercial interest in multilingual products. Furthermore, proprietary packages largely use inflexible closed standards that do not make it easy for users to add their own enhancements. By contrast, a local NGO called Translate (http://www.translate.org.za/) has already released a Xhosa version of some applications after only 3 months of work.

They were successful mainly because they used KDE, an OSS desktop environment akin to Microsoft Windows in functionality. KDE itself is sensitive to human language issues and is currently translated into 42 languages, far in excess of any of the popular proprietary packages. It took Translate six weeks to translate enough of KDE into Xhosa to make it ready for release. Another six weeks were spent for other minor components and documentation. It proved so easy to add Xhosa to KDE because of the spirit of co-operation and collaboration in OSS projects. As a result it enjoys some of the richest translation tools and is multi-lingual from the ground up. Information is freely discussed and shared, which means that KDE, like most successful OSS products, is rapidly being enhanced by thousands of volunteer programmers around the world.

[42] See "http://www.zdnet.com/zdnn/stories/news/0,4586,2781914,00.html"

Some effort is required into translating tutorials, help files and other content (often omitted in exercises of translating application user interfaces).

Finally, within the ICT sector, Southern African capability in software development (customising and enhancing open source software), systems administration and support need attention. Although these groups tend to be more independent and self sufficient, localised on-line content and community building would be invaluable, as would any efforts to lower barriers (such as currency exchange rates) to certification.

### *B. The Role of FLOSS in Education and Innovation Systems*

Education systems around the world are experiencing a range of drivers for fundamental change, both from within and without. These include:

- globalisation;
- changing concepts around the role of knowledge, knowledge workers, knowledge citizens, innovation systems and learning organisations;
- the widespread need for quality life-long learning; and
- the relentless emergence of new ICTs, coupled with their growing penetration of, and impact on, all sectors of society, including the most disadvantaged.

Understanding of the role of education has broadened significantly, recognising that learners need to think independently and critically, and collaborate with others to make sense of their changing environment. In the developing world in general, and Africa in particular, the education systems are experiencing the greatest pressures to transform, but have the poorest resources with which to respond. The imbalances between the global supply of education and the magnitude of the demand for access make the imperative of providing a decent “education for all” (EFA) one of the greatest moral challenges of our age.

**BOX 4: Examples of OSS Applications: Rural Education**

A Cape Town-based company has used Linux to come to the rescue of a rural school in KwaZulu Natal with a pilot project to deliver Internet content where there is little reliable telephone connectivity. Wizzy Digital Courier, which delivers Internet content via a courier system, and offers cost effective computing and Internet access to schools, has launched a pilot project in Eshowe, a small, rural town.

KwaZibonele Junior Primary - a school with 950 pupils in grades one to four and with an average age of nine - was selected for the pilot. The school had previously been donated eight computers but had little use for them. Wizzy Digital Courier has installed two servers: one used for e-mail and radio connectivity, and a Linux thin-client server to replace the Microsoft software on the school computers. The thin-client software reduces maintenance because instead of performing maintenance on every computer in the network, all configuration and software installation is performed only once on the thin-client server.

Students have already moved from learning to log-in to basic computer literacy skills, mouse practice, drag-n-drop, and keyboard familiarity by means of Linux programs that are fun and educational,.

Wizzy Digital Courier has also secured a contract with Eshowe High School to refit its computer room using the system deployed at KwaZibonele Junior Primary. Eshowe High, an ex-Model C school, has 620 pupils.

The situation in South Africa becomes even more concerning when the teaching of ICTs is considered. Currently, available ICT related courses provide a range of accredited qualifications including degrees (graduate and post-graduate), diplomas, and certificates. These are primarily aimed at creating technicians, technologists and professionals for the long established ICT sector and their traditional markets (e.g. military, finance, retail, etc.). These are important people, and Africa has an immediate need for many more of them, and of world class calibre. However, there is a much broader range of ICT capabilities called for to satisfy the growing needs of Africa's emerging knowledge economy. This is a chicken and egg situation - if we do not provide such people, society will not grow, i.e. it cannot be demand led. Our society must, collectively, take the risk of anticipating (and thereby stimulating) the demand for local expertise.

Currently, there is significant evidence that growth of South Africa's ICT industry (and particularly its indigenous component) is being seriously stunted by the insufficient supply of appropriately qualified, trained and/or experienced people. One of the major roles of HEIs is the output of quality graduates who are useful to industry and society, and who are equipped with the appropriate types of skills. The recent ISETT SETA study estimates that up to 165 000 students will be needed to address ICT skills shortages in the near future. Estimates of numbers of ICT professionals in South Africa presently range between 54000[43] and 74500[44], while Forge-Ahead BMI-TechKnowledge's survey of black ICT companies and professionals indicates that in

[43] International Labour Organisation (2001).
[44] ISETT SETA (2002).

**Box 5: OSS Parables: Part I - Funeka's awakening**

Funeka is a schoolteacher with a mission: to give her dusty, rural school the very best. She launches a campaign to build a computer lab and approaches various businesses for help. To her delight, one company donates 20 computers that are being replaced, but the company will keep all their software licenses for their new machines. She also has to find her own educational software.

Delight turns to horror when she discovers that it will cost many thousands of Rand for software licenses, including licensing the educational software the dealer tells her she needs. To make matters worse, casual inspection reveals that the content is geared to American schools, using unfamiliar baseball metaphors and the like.

Meantime, Funeka's students have been doing some legwork of their own. They have contacted a young IT company that has offered to network the computers and connect them to the Internet. When the company's network guru calls by and finds computers with no software, she installs Linux and associated free software on all of them, sets up the network and Internet connection and even gives the students a preliminary driving lesson on using the software and surfing the Internet.

While Funeka agonises over raising a software budget, the students spend many days probing, exploring and discovering new things. Within a short time they have learned to do creative projects by searching the Internet and sending email around the world for facts they can't find in the tiny school library. Using tools and examples from other Web sites, they soon start designing their own school Web site and developing content like a Web-based newspaper covering school and local community issues.

When she learns of all this, Funeka is amazed at the creativity of her students, and decides that her original idea of what computers should do is completely wrong. She had thought of the computer as just another passive medium of instruction.

Funeka quickly adapts to this awakening, and promptly arranges a session on the Internet - given by her students to members of staff. They are all amazed that all this has happened without the school having to pay a cent in software licenses.

They also heartily approve when the students explain their plans to design a community resource for guided access to government Web sites. The one concern the students have is that they are often unable to read files downloaded from government sites. The problematic files are in a format that requires proprietary software to read.

**DISCUSSION: Funeka' story demonstrates the need to think clearly about the problem one is intending to address. In the school context, self-initiated exploration, formal instruction and school administration are three distinct areas. Think carefully before making a huge investment in expensive software. It also touches on the importance of promoting access to government information through the use of non-proprietary data formats.**

2000 only about 5000 of them were black. A cause for serious concern is that no convergence is anticipated between supply and demand during the period ending in 2009.

A recent study undertaken by the Department of Trade and Industry, which examined the rate of diffusion of ICT within eight industry sectors, showed that HEIs were hardly considered by industry as sources of information or training in ICTs (Esselaar et al. 2002). The reliance instead on vendors for learning material about ICTs is of great concern since such material is often vendor-specific rather than generic.

The fundamental message is that South Africa's current education system, given the pressures described above, and compounded by the dynamism and expansion of the ICT 'realm', cannot alone even begin to satisfy the wide range of training and education needs (for ICT in general, and FLOSS in particular) that are required in the immediate future. Hence, there needs to be a national strategy to address this special need, otherwise the full potential of FLOSS will not be achieved.

**BOX 6: OSS Parables: Part II - Sipho's choice**

Sipho has good reason to be pleased with himself; he has just submitted a groundbreaking PhD thesis at a leading South African university. Using advanced concepts in mathematics and physics, his thesis, "QVM: the Quantum Virtual Machine", proposes an ingenious algorithm to speed up the conventional PC beyond the wildest dreams of classical wisdom.

QVM will make light of computer resource hungry fields like environmental and climate modelling, determination of protein structure and function, discovery of new drugs, complex industrial simulation and design etc. It will also lead to a host of completely new applications that inevitably accompany such a major computational advance.

Sipho cannot wait to publish a paper in a high impact international journal giving full details of QVM principles and design. He also intends to place a full software implementation on the Internet, allowing anyone to download and use it on a standard PC. No license fee, no royalties. They can use the software as they please –learn from it, modify it - as long as they do not repackage and sell it for private commercial gain and attempt to stop others from using the free distribution.

His friends are horrified - he could license QVM to a global computer company and make a fortune. The university is horrified - it could license QVM to a global computer company and make a fortune. His supervisor is horrified…

But Sipho stands his ground. He firmly believes in the freedom (or should that be obligation?) to publish academic work supported by public funds – software included. His own research benefited immensely from the use of software distributed under similar conditions.

He is also mindful of a moral obligation to seek the greatest economic gain for the country from publicly funded research. But this only strengthens his resolve. He is convinced that greater benefit can accrue to South Africa's scientific and economic fortunes through his suggested route than by surrendering such a major scientific breakthrough wholesale to any single company, whether it is foreign (almost certainly) or local.

"Is he very foolish or simply ahead of the game, like he is in his research?" his friends puzzle. "Is he really acting in the country's best interest or is he a well-meaning but naïve academic?" wonders the inquiring public. "Should a man like this even be allowed a choice on the matter?" fumes the university's deputy vice chancellor for research.

DISCUSSION: It is one thing to make use of currently available open software, but when should software developed with public funds be open source? Whose judgement call should it be? Sipho's story raises this question and a number of related issues.

ICT training is essential for South Africa's public and private sectors to grow their capacity for service, product innovation and improved efficiencies. There is a growing belief that the extensive inclusion of FLOSS at all levels of ICT education has become essential in South Africa's current circumstances, and that HEIs should do more in using FLOSS to produce graduates with greater objectivity and understanding.

There are three main areas of ICT & FLOSS teaching/learning/awareness that should be considered:

- *ICT and FLOSS Literacy/Fluency:* Most current literacy material is outdated (e.g. 'computyping'), and the dynamism of the ICT sector demands updating at least annually. In addition, much of the material has not been designed for the many thousands of potential learners from rural and remote areas. In such cases, it is essential that, for example, language and cultural issues are catered for if we are serious about bridging the 'digital divide'. Hence, improved ICT and FLOSS literacy learning material that satisfies these needs should be distributed on an ongoing basis.
- *FLOSS Benefits Awareness:* This needs to be created for all members of the leadership corps of (South) African society, including the public sector, private sector, civil society, academia, rural communities, etc. The emphasis of such

material should not be on the technologies themselves, but on the potential impact (especially benefits) of FLOSS, now and in the future, tailored to the areas of society most relevant to each leader. Clearly, this cannot best be addressed by any sector (ICT, tertiary, public, etc.) acting in isolation.

- *FLOSS Technologists and Professionals:* The theoretical content needs in this area for the ICT sector are being addressed by the current courses and material of many existing HEIs. The major concern from the private sector is that far from enough appropriately qualified people are being produced, especially from the black community and women, and concerns are also voiced that many course components are permanently outdated (due to the years currently required to rewrite curricula contrasted with the dynamism of ICTs). What is not being addressed, however, is the large and growing need for such FLOSS technologists and professionals in *all other* sectors of society, (private, public, development, etc.). These people need to be educated to comprehensively understand not only their sector (e.g. agriculture, tourism, etc) but also all aspects of ICT and FLOSS relevant to that sector (current and future).

Note that there are a number of initiatives addressing some of these needs already in Southern Africa. Examples include the activities of The Shuttleworth Foundation, Schoolnet, HP and the CSIR Open Source Centre. The challenge is to leverage the synergies among these efforts to make a real difference in the region.

### *C. The Private Sector*

An enterprise that is considering the adoption of a FLOSS solution for its ICT needs, should consider the relative advantages and disadvantages of the available open and proprietary solutions.

**Advantages of FLOSS for users and usability status**

***Pricing:*** While, for large companies, the cost of software licensing is not a major proportion of the total costs of ownership of ICT, the relatively low price of FLOSS products allows enterprises to try the product and facilitates its rapid diffusion. Users are also able to avoid being locked in to buying future upgrades and can avoid to a significant degree the management costs of licenses.

***Robustness:*** In a FLOSS development model like that used in GNU/Linux, peer review leads to a community of participants who help debug the software and make it more resource-efficient. In contrast to proprietary software where testers must be employed, FLOSS typically depends on global testing which leads to a robustness through the spectrum of conditions experienced where the software is deployed.

The security of FLOSS is also a product of this robustness in development, with the structure of peer review acting to increase the detection of security flaws rather than efforts to conceal software defects.

**BOX 7: Examples of OSS Applications: LegalWise Internet Server**

The LegalWise company had unique requirements for a general-purpose solution. Although they were relatively low-traffic Internet users, they needed a cost-efficient means for providing access to their branches around the country. They also needed a means of managing local e-mail accounts on local servers while still managing mail transfers countrywide.

The Linux Internet Network Connection (LINC) server solution proved ideal because of its low cost and reliability. The LINC server has made it easy for LegalWise to configure a nationwide point of presence where remote branches can find their way through to their local domain. Everything is now working far better than before - and at a lower cost. Since the LINC server has no licensing fees, LegalWise can add capacity without incurring additional software charges.

A Linux specialist emphasised that many businesses today are struggling with the onerous costs of licensing proprietary solutions. This Linux solution provides all of the above functionality without the need for constant upgrades or annual license fees. For mission-critical situations the Linux solution is the best option with its ability to handle high traffic situations and its advanced anti-spam capabilities. The server includes a complete firewall with security alerts to warn of any attempts to compromise security.

***Flexibility in Use:*** The update cycle of FLOSS can often be much more responsive to the users' requirements than proprietary software. As FLOSS is typically written portably, it is usually available for use on a wide range of platforms. The openness of the source code also allows for the development of portability to other platforms, as the user requires.

***Reduction in duplication:*** Large fragmented organisations are striving more and more to reduce costly and ineffective duplications of ICT functionality (often via differing proprietary solutions) by implementing collaborative team based cross-organisational ICT projects. FLOSS has been found to lend itself much more naturally to this process.

**BOX 8: Examples of OSS Applications: Lewis Stores**

Lewis Stores operates 460 retail stores in South Africa and neighbouring countries. In 1998, a decision was taken to adopt a software point of sales solution in all its stores based on the Linux operating system. This OSS solution caters for all the transactions processing at store level for the management of debtors, stock and cash. To achieve this, the company selected Universal Computer Services (UCS) as its software development partner.

Currently in 2003, the system continues to provide the consistency and stability required of a decentralized processing environment. Each day the stores transmit their transactions to Head Office where the consolidation takes place.

Since first deploying its retail application on Linux in 1993, UCS has established a significant Linux presence across Southern Africa. It has deployed its Linux based systems to over 4,000 stores (equating to 20,000 Linux workstations) across a number of internationally represented corporate retailers. Very recently, UCS' innovative J2EE large-scale retail application, wholly developed and supported by South African resources, won the company the prestigious TT100 award.

All UCS products are developed entirely using OSS and are license free. From the J2EE (Java 2 Enterprise Edition) web-based architecture, to the Linux operating system, ISQL complaint store databases and web, application and messaging applications, UCS makes no use of licensed 3rd party applications. This ensures negligible, if any, platform costs or annual platform maintenance fees, thereby guaranteeing predictable and low cost of ownership.

***Increased competition:*** By freeing organisations from vendor lock-in, the use of FLOSS, open standards and open content stimulates competition, often by allowing local players into a previously closed market. In addition, through embedded open standards in FLOSS, this leads to greater interoperability.

***Users' Independence from Suppliers:*** FLOSS does not have a single source for support and upgrades. This reduces the risk experienced with proprietary software that the company will stop supporting it or go out of business. It also allows a competitive market to develop in support services.

***Proliferation of GNU/Linux Distributions:*** The distributions package collections of popular FLOSS products orientated towards the needs of a typical client base. Users may therefore select a suitable distribution closest to their needs in contrast to the proprietary approach which often seeks to be all things to everyone, providing superfluous components and a high processing overhead.

***Ease of Use:*** The history of the FLOSS development model is dominated by technical developers/users rather than end users. Hence "user friendly" features (such as graphic user interfaces) have been more recent developments as the popularity of FLOSS solutions has spread way beyond the initial technical 'clique'. Currently, the usability of FLOSS solutions is more than adequate for end-users. Avoid the mistake of considering an operating system usable only if its usability is better than that of Windows in every way.

***Marketing and Support:*** Unlike proprietary systems, there is not a single owner organisation dedicated to supporting each FLOSS product. Hence, FLOSS product awareness tends to be neglected. It is important to note that these gaps are being filled by the cooperative development communities, professionals and even proprietary vendors deploying their products on GNU/Linux.

The points above are relevant to the application of FLOSS in the private sector, regardless of the broader socio-economic environment. However, private sector

utilisation of FLOSS in a relatively less economically developed setting involves some important additional issues which are discussed next.

**BOX 9: Examples of OSS Applications: Nando's and Linux**

Nando's, a customer-orientated business, cannot allow its services to be limited by ICTs: the familiar excuse "Sorry, our computer system is down" is not acceptable. On this basis, they have defined six critical elements for their ICT systems, which only Linux (which they have been using for the past 9 years) could satisfy:

***Reliability:*** Linux has proven exceptionally reliable in both the retail stores (very hostile environments for PC's) and on the Head Office servers. The Back Office systems seldom, if ever, need to be rebooted due to operating system problems – only power related issues.

***Stability:*** Stability of Linux is beyond comparison, with an automatic reboot being scheduled every two weeks to kill any stray processes. Other than that, they are almost never required.

***Security:*** Linux can be configured to restrict inappropriate access by general users, e.g. to install games and other unofficial applications. Access to the system by "hackers" is far more difficult than with other operating systems.

***Scalability:*** This appears almost unlimited. Our developers, Obsidian Systems, have easily satisfied our sometimes unusual requests, e.g.:

- They quickly wrote a routine that, every 5 minutes, monitors the status of our 11 diginet links to Regional Offices around the country. Any faults are reported by SMS to a support person's mobile phone, and a call is logged with Telkom before the user even knows of the problem.
- They easily wrote a script to scan specific ports on our Linux firewall and report the IP addresses of users' machines (using a proprietary OS) that were infected with the Blaster and Welchia viruses.

***Flexibility:*** Linux will fit into, around, over or under any other OS. Products such as Samba, SSH allow for seamless integration, data extraction etc with other Operating systems.

***Communications:*** Linux communications are unique, and can be fully customized and scheduled. We currently extract data from 180 remote sites without any human intervention. Failures are caused by unpaid phone lines or hardware problems. Our international operational team access their mail anywhere on the planet, using Linux Firewall and Linux Mail Server.

Nando's believe that OSS has given them a competitive edge, and are planning more extensive developments in the near future, including:

- Migrating to a Linux data-warehouse from their current proprietary system.
- Re-developing their back office systems using OSS.

### *How Indigenous is South Africa's ICT Industry?*

Over the past three decades, the convergence (both in terms of technology and markets) of telecommunications, the mass media, networked computing environments, and the Internet has changed the way the developed world works and plays. This ICT based "network society" is seen as the generator of a "new economy", manifested in such icons as Silicon Valley and the Asian Tigers. However, almost all of these dramatic changes have been taking place in the developed world, whilst the developing world, and especially Africa, appears to be falling ever further behind.

To date, ICT industries have not been major players in the economies of most African countries, which tend to be consumers (and only in large organisations and elites), rather than producers to any significant degree. Africa, led by South Africa, has been importing developed world ICTs and employing them in a variety of ways for more than two decades, but with negligible impact on the inclusion and poverty alleviation for the vast majority of its people. South Africa has the largest ICT Industry in Africa. In 2002, it was the 20th largest country market for ICT products and services, accounting for 0.6% of worldwide revenues (estimated in 2000 at R79 billion). Software accounted for only 18% of ICT expenditure in 2000, whilst hardware purchasing accounts for 46% and services a rapidly growing 36%. But is this an indigenous ICT industry?

**BOX 10: Examples of OSS Applications: Document Management in the Medical Research Council**

Cape Town-based software development company Jam Warehouse has released an open source enterprise document management system. Called KnowledgeTree, it is based on an application developed in collaboration with CS Holdings for the Medical Research Council of South Africa. The product builds on the company's experience of content and document management systems for clients like Independent Newspapers South Africa and UK retail giant Tesco Stores.

The project began with a rather basic OSS solution. The MRC has funded its further development into an enterprise document management system in order to manage their knowledge assets and will benefit from future improvements introduced by other organisations. In this way, the MRC has not only provided itself with an independent, strategic solution, but it has also exhibited the vision to support the development of an indigenous South African software industry, as well as aligning with the government's OSS Strategy

Although many organisations are wary of OSS applications, Jam Warehouse has found that careful project and quality procedures make OSS products as reliable as anything else in the market. "There is a misconception, particularly in the South African market, that OSS is inferior. In our experience, OSS applications are as secure and scalable as those built on proprietary platforms. We have recently released large applications on both Microsoft and OSS platforms. Both environments have their advantages."

The corridor between Johannesburg and Pretoria can be described as an ICT cluster, centred on Midrand. But very few of the ICT products are South African in origin. The local ICT industry is made up of outposts (satellites, or fully dependent subsidiaries) of mostly multinational companies who find Midrand the best environment to sell on their products, developed in the USA or Europe, primarily into the developed component of the South African market, but also into other African markets.

In the rest of Africa, much of the money used to purchase and maintain large ICT applications stems from donor funds. Hence, donor funds, which superficially appear to be 'invested' in Africa, are often used to buy developed world applications for installation in African government premises. The foreign reserves immediately flow

**BOX 11: Examples of OSS Applications: FotoFirst Data Management**

The Teltron group, owners of FotoFirst and other retail photography businesses, has turned to Linux to manage its software and data synchronisation across its more than 80 branches in South Africa, Namibia and Botswana. The company has deployed Obsidian Systems' Synchrony solution to manage its network of point of sale terminals.

Each Teltron branch runs a Linux-based automated point of sale system which returns daily transactional information back to the head office's financial system. Teltron's previous system was manual. Drivers would have to visit branches and bring back physical records, which were then captured at head office.

The system ensures that all nodes are running the newest version of the client software, and all local databases are 'in sync' with centrally shared databases. This means that all software issues are resolved from a central point, as changes are spawned to every branch. This, in turn, ensures total synchronicity throughout the organisation. Teltron's software and scripts can be updated or changed remotely on a completely automated basis, ensuring that the company has complete control over each branch's software from a central location.

The system has a much lower total cost of ownership for Teltron. Each branch's system runs on a tamper-proof customised server. Using Linux ensures that Teltron has no license fees to worry about. There is very little scope for error since update processes are controlled by head office and no staff members are required on the client side to perform updates manually.

back to the developed world, without circulating in the African economy, and the potential impact of the ICT 'solution' compromised, because it is a solution developed for a different problem, involving different people with different organisational circumstances and culture.

Many believe that ICTs, appropriately approached by the whole of (South) African society, can dramatically affect economic growth and poverty alleviation throughout Africa. Manuel Castells (and others) state that *"the next phase (of the information society) is the scientific convergence of information technology/media and social sciences (psychology, sociology, economics, etc.)"*, and that *"the current trends in the expansion of the information society, worldwide, point toward the increasing importance of the social uses of ICTs"* (Castells and Himanen, 2002). This poses an exciting prospect for (South) Africa.

**BOX 12: Examples of OSS Applications: SME Opportunities - Kgatelopele Technologies**

Kgatelopele Technologies was founded in 2002 by Thabo Mogaswa, a Software Engineer who found the benefits of OSS too convincing to pass up. The company focuses on developing software solutions for a wide range of customers using OSS.

In May 2002, the company started working on a School Management Tool, using OSS, that meets the needs of township schools in South Africa. The solution was named eSchoolMan and can be used by schools with limited and/or old ICT infrastructure, which fits the description of most schools in South Africa and Africa.

Previously, these schools relied on capturing all information manually on registers and schedules. This proved time-consuming both at the beginning of the year (when student registrations, new intakes, and class and subject allocations were captured) and after tests and exams (when student marks were captured). eSchoolMan was developed to improve the efficiency and effectiveness of school information systems used in South Africa, and was well received in the schools used to pilot the solution. It became the basis for the Open Source School Management System (OSSMS) Project, funded by DFID, which was aimed at developing an enhanced version of the tool, suitable for use throughout the Southern African Region.

Kgatelopele has also used OSS to customize world-wide solutions for specific customer needs during the GITO Council Portal Development Project. Using an existing open source Content Management program, the company was able to incorporate three indigenous/South African languages, in addition to English. That project marked the beginning of many other OSS-based solutions, especially for Government, to deliver services to South Africans in a language of their choice.

Kgatelopele generated over R600 000 in revenues in its first year of business, confirming the real opportunities presented by OSS to other SME's, local software development companies and young entrepreneurs to develop and stimulate an indigenous ICT industry. Kgatelopele Technologies views OSS as providing them with a real competitive advantage.

It suggests that by focussing much more effort on applying ICTs to satisfy the needs of the poorest and most remote areas, Africa is not only likely to accelerate development on a broader scale, but also it is likely to emerge at the forefront of the social uses of ICTs. In other words, Africa could develop indigenous ICT industries which primarily address their own development needs (which imported developed world ICTs have proved unsuitable for over the past 15 years), but which subsequently could emerge as leaders in the next phase of the global information economy – i.e. "ICTs with a soul".

This suggests a path for the development of indigenous ICT industries in (South) Africa. The people who understand these development needs are those from the local, disadvantaged communities, not those sitting in offices in Midrand, or San Jose. The organisations that are best fit to address these needs are small, local companies (individuals or start-ups) – i.e. the SMME sector, NOT the multinationals. There is a wide range of development needs (explicit and implicit) in most disadvantaged communities; hence, there is a large set of potential solutions and markets. But there is not much investment money available in these areas. It is unlikely that any proprietary ICT vendors (particularly multi-nationals) would seriously consider many, if any, of the innovative ideas from this under-developed, seemingly unattractive market. However, by adopting FLOSS, the local entrepreneurs and champions are immediately

empowered to pilot their ideas themselves, with very low start-up costs. They do not have to reinvent wheels, but can adapt and build on what is already freely available from the global FLOSS community. The biggest remaining obstacle is the lack of appropriate FLOSS training and skills amongst the most disadvantaged communities. That is a problem which (South) Africa can overcome, if approached strategically with innovative methods (see boxes 3, 4, 10, 11, and 12).

# IX The Road Ahead

This document is an evolving work in progress. It is built upon the original document in which several recommendations were made regarding FLOSS, with particular emphasis on the public sector. Given the time that has passed since these recommendations were proposed this section briefly reviews what efforts have been made in meeting them. It concludes with suggestions regarding possible paths forward for FLOSS in South Africa.

### *A. What has been achieved?*

Table 1 lists recommendations (directed primarily towards utilisation of FLOSS in South Africa's public sector) stated in the first version of this document (2002).

| | **TABLE 1: Previous Recommendations Regarding FLOSS** |
|---|---|
| | **ICT in the Public Sector:** |
| 1 | Make Open Standards a non-negotiable base for ICT in the Public Sector. |
| 2 | Encourage government agencies and public institutions to use Open Software whenever feasible. |
| 3 | Allow Open Software to compete on a "level playing field" with proprietary alternatives in government software procurement. |
| | **Open Software Development** |
| 4 | Promote documentation, translation and localisation of software, especially for use in the Public Sector. |
| 5 | Promote Open Software in pre-commercial research and development projects financed with public funds. |
| 6 | Establish an Open Software Development Initiative. |
| | **Training and Support** |
| 7 | Promote education and training on Open Software products. One aspect of this is a general education that lays emphasis on principles rather than specific software products. The other aspect is the shortage of trained people to use and support Open Software solutions. |
| 8 | Establish a national capability for testing, evaluation, verification and accreditation of Open Software. |
| 9 | Provide incentives for Open Software training and development. |
| | **Legal Issues** |
| 10 | Oppose patenting of standards, software and algorithms. |

Since then the South African government has made great strides in adopting its proposed ideas and recommendations. This is illustrated by the following important documents that have been produced in the interim:

- *January 2003: "Using Open source software in the South African Government".* This document, produced by GITOC, contains policy recommendations and implementation targets, and represents the current accepted policy of the South African government.
- *May 2003: "Designing and managing a framework for assessing results of use to OSS in South Africa: Phase 1".* This document, produced by SITA, also supports the FLOSS implementation process.
- *November 2003:* The Presidential National Commission on Information Society and Development commissioned a study on South Africa's OSS policy. It is expected that this document will improve on the current OSS policy and provide guidelines for a move to the next phase of its implementation.

### *B. Mobilisation and engagement*

Version 1 of this document emphasised the public sector in its recommendations regarding use of FLOSS. This version broadens and extends the recommendations to include public and private sectors, with specific recommendations for the environment, agriculture and education domains. These are summarised in Table 2.

A troubling feature of the operations of some multi-national corporations is that they are often major absorbers of public sector and donor funds. While these multi-nationals can contribute to the development of indigenous capacities in ICTs, these contributions are inevitably less significant than local efforts because:

- their focus is limited to demand from the most economically developed segments of society, and
- the capabilities developed tend to be vendor specific and limited in scope.

Increasing the human capital stocks trained in FLOSS is fundamental to creating a general and truly sustainable advance in (South) Africa's ICT industry:
Firstly, the vast training effort required at several skill levels (e.g. end user, user support, developer, systems administrator) could stimulate a local FLOSS/ICT training industry, which could evolve as the maturity of the trainees develops.
Secondly, many of those trained are likely to begin to produce content relevant to African communities as well as a range of services to handle, disseminate and grow this flood of locally relevant information (see Box 3).
Thirdly, a smaller number of those trained in FLOSS should begin to build locally relevant tools and applications, helping to improve FLOSS/ICT literacy even further and opening up more services at the local level (see Box 12).

Funding and investments by donors and the public sector on such locally, FLOSS-based content would have a greater circulation within the local communities and increase the potential to generate a 'wealth-creating' spiral likely to spin off a wide range of local services, many of which might not involve ICTs. Further, this would be likely to decrease the isolation of these communities from each other, the public sector and the broader global society. The durability and sustainability of such a vision requires that involvement expands to all segments of society.

### *C. Next Steps – Beyond the Public Sector*

Government has taken an insightful lead in adopting FLOSS, as well as developing policies and strategies that are moving from 'levelling the playing field' for FLOSS, to a more pro-active approach in recognition of the need to overcome the entrenched (market positioning) advantages that proprietary systems have inherited. There is a long way to go, particularly in coordinated implementation at the departmental level, but government is clearly on the right path. However, should it be only the public sector which responds to the opportunities being presented by FLOSS? What should happen in the rest of South African (and African) society? By considering several other sectors, a range of questions arise.

***Large Private Sector Organisations:***

Most large South African organisations have been employing ICTs for years, but the vast majority have invested significantly only in proprietary systems. Yet, in contrast:

- Why, since 1995, has GNU/Linux proven exceptionally reliable for Nando's, both in their retail stores and on their Head Office servers (see Box 9)?
- Why, since 1998, has Lewis Stores successfully operated a decentralised point of sales processing environment across its 460 retail stores in South Africa and neighbouring countries based on the GNU/Linux operating system (see Box 8)?

We are arguing that these examples could be much more widely replicated throughout South Africa (and Africa), with benefits not only to the organisations themselves, but also to the broader South African economy. To achieve this, hasn't the time come for every large private sector organisation, at the very least, to objectively consider the benefits they would experience if they were to migrate to FLOSS in the near future? But, who can help them turn the generic list of benefits that are available from a variety of sources (including the Web and documents like this) into meaningful strategies and objectives in the particular context of their business practices and market area? How well documented is the migration to FLOSS, and which organisations exist that can provide quality migration services? In particular, which of these large private sector organisations is going to migrate first, and why?

***Small Private Sector (SMMEs):***

FLOSS presents significant opportunities and benefits for SMMEs.

By considering FLOSS alternatives to the currently dominant proprietary systems, many more SMMEs could afford to employ ICTs, the benefits of which would not only be experienced by themselves and their customers, but, via a multiplier effect, could have an impact on the national economy.

The challenge is to ensure wide access to the various resources that are available to facilitate and enable SMME innovation and sustainability. Perhaps the answer to this challenge points towards a major opportunity for the establishment of a new wave of SMMEs to provide such services, thereby initiating a truly *indigenous ICT industry*.

Indeed, there are some success stories to show that this is already happening:

- A black start-up company, founded in Pretoria in 2002, generated over R600,000 in revenues in its first year of business. It focuses on developing FLOSS solutions for a range of local customers, including:
    - An FLOSS-based school management tool meeting the needs of township schools with limited or old ICT infrastructure.
    - Incorporating three indigenous/South African languages into an existing FLOSS content management programme (see Box 12).
- A Cape Town-based software development company has released a FLOSS enterprise document management system based on an application developed for the Medical Research Council of South Africa. The MRC has not only provided itself with an independent, strategic solution, but it has also exhibited the vision to support the development of an indigenous South African software industry (see Box 10).

How can the efforts of the government, local and international private sector organisations, as well as the international community be coordinated and focused into quickly converting these encouraging but rare examples into a healthy flow of sustainable growth?

***Disadvantaged and Rural Communities:***

A wide range of initiatives are currently attempting to provide ICT access, literacy and connectivity to the poorest communities (from public sector, international development community, NGOs, etc.). However, comparatively few are implementing FLOSS options, despite the dramatic savings, longevity and better reliability of, for example Linux labs. Although all the relevant information is freely available, it is seldom being seen or heard by the effected decision makers and stakeholders. How can FLOSS be established as an essential component of all ICT for development initiatives, particularly for the sustainable benefit of the poor?

- Can culture and language issues be effectively and widely addressed via anything other than local capability and capacity utilising FLOSS?
- What impact could FLOSS have in providing primary health care services out to the very poorest and most remote communities?
- Opinion is growing among development experts that ***all*** sectors of society (but particularly out of school youth and unemployed adults) need to be enabled via a broader concept of literacy, which includes information and ICT literacy. What should the FLOSS component of this broader literacy be?

***Environment and Agriculture:***

FLOSS is already playing a significant role in several tools used by these two, related sectors which are so important in the sustainable development of South Africa and Africa. However, many institutions in developing countries continue to use expensive proprietary systems. How can the awareness of the alternatives be spread, and the development of the existing FLOSS systems be accelerated? There is a clear need to stimulate and facilitate communication and coordination in these sectors, assess common needs and integrate existing FLOSS offerings for the common good.

***The International Development Community:***

Some global organisations (e.g. UNDP, UNESCO) are already promoting FLOSS in many of their ICT-related activities. However, many other donors and foundations tend to promote proprietary solutions as components of their contributions to individual and consortium projects. How can South Africa and like-minded influential developing countries (e.g. Brazil, India and China) work with UNESCO, UNDP and other global institutions in the pro-active development of de-facto, open, ICT for development standards?

***National Systems of Innovation:***

Globalisation and the emerging knowledge economy are presenting challenges to countries across the planet, whether industrialised or under-developed. There is compelling evidence (Castells and Himanen, 2002) that the countries which are adapting best to this new world in terms of growth, improved quality of life and reduction of poverty are those with the most effective systems of innovation, involving close

collaboration between the public sector, the tertiary sector and the private sector. In the developing world, there are few, if any, nations where these systems of innovation are in place. South Africa has recognised this problem for some time, and has undertaken several national initiatives to help establish an effective national system of innovation. However, do we fully understand the role FLOSS has played in innovation, especially in the world's most innovative societies, over the past 40 years? Should we not investigate, customise for the South African context, and develop appropriate FLOSS-based innovation for development models?

| TABLE 2: Further Recommendations Regarding FLOSS Utilisation |
|---|
| **Issues to Take Forward** |
| |
| **General Actions and Best Practices** |
| Proactively avoid repeating history (by following the other recommendations). Ensure we become active participants in the FLOSS movement and don't miss the boat. |
| Raise awareness of the benefits of FLOSS: targeted road shows by the FLOSS communities. Stimulate and support adoption and use. |
| |
| **Coordinating Activites** |
| Target the International Development Community. Make a concerted, coordinated effort to raise their awareness and understanding of the FLOSS movement, open standards issues and potential synergies. Coordinate all ICT for Development funding for a consistent message, to optimise resources, and maximise collaboration opportunities. |
| Support current common donor activities: such as DFID's project to draw up guidelines for development project prioritisation, the OSISA Funders' Forum, etc. |
| Funding challenge: prioritising and selecting projects for support. Research on guidelines and best practices for studies of feasibility, sustainability and impact. Understand the multi-factorial systemic inter-relations among social, economic, ICTs in general, FLOSS and environmental factors. |
| Coordinate the open source software development communities to be able to tender for large integrated projects. |
| Public sectors cooperate with other nations to develop standards and build solutions of mutual interest. |
| |
| **Policy/philosophy** |
| Choose FLOSS and open standards over proprietary. |
| Raise the debate on Free and Open Source Software to ensure relevant parties really understand the implications of the broader social phenomenon. |
| Recognise the link between right of access to information and FLOSS (refer to the Peru-Microsoft letters - Appendix 3). |
| |
| **Software Development** |
| Promote the hybrid model of software development: FLOSS projects with contributors around the globe. Local developers (often in agile co-located teams) paid on a needs-driven basis to build solutions to meet development requirements and goals. In the process, these teams contribute to and initiate international FLOSS projects. |
| Adopt an Open Standards based approach in ICTs to facilitate collaborative development among distributed teams and reuse of components. |
| |
| **Meeting Development Needs** |
| Identify and prioritise local needs for sustainable development and start processes that will start with idea generation and end in sustainable solutions (e.g. well-chosen public funded projects). |
| Research approaches to assessing broader societal impacts of FLOSS. Indicators, sustainability, innovation. |
| |
| **IPR** |
| Take account of organisational and national business climates when considering FLOSS vs OSS vs PS. Sensitivity and flexibility. |
| Oppose software patents in the face of a strong international trend of strengthening of IPR. |
| Update SA legislation in an informed manner. |
| Raise awareness and understanding of the debate, and instigate use of Creative commons licenses for open content where applicable. |

| |
|---|
| **Skills, capacity building and innovation** |
| Study the role of FLOSS in innovation internationally. Understand how it works in the most innovative countries around the world. Apply and adapt the principles for local conditions. |
| Target national systems of innovation in a coherent fashion. This includes TEIs, facilitating collaboration among public, tertiary and private sectors. Stimulate the knowledge economy. |
| Study the success stories across sectors and levels within them (e.g. of government) and share the knowledge to produce a healthy flow of sustainable growth. |
| Emphasise the importance of open content for learning ICT and other subject matter including basic literacy. Link up with and learn from existing initiatives. |
| Augment the technical ICT education in TEIs with domain and sector related ICT training. For example encourage the Learnerships to include ICT literacy with open source software. |
| Understand the local needs and issues in less economically developed conditions and how socio-economic factors interact with technology interventions, ICT adoption and development. The DIY approach seems most effective. |
| Recognise the innovation potential of rural communities and enable the innovation by providing access and support towards sustainable systems of local innovation. |
| Identify the real educational needs and which gaps may be filled by FLOSS. |
| Training and Support: build capacity and stimulate SMMEs founded upon FLOSS: support, installation, customisation, training, and software development. |
| |
| **Public/National/Civil Society** |
| Promote South Africa as a pioneer and model for FLOSS development in the South. |
| FLOSS as part of an effort to ensure access to public records and meeting the right of citizens to have such access. |
| Consider FLOSS as an essential component of any ICT for Development initiative. |
| Raise awareness of the benefits of FLOSS to the point of action among decision makers and stakeholders. |
| |
| **Private Sector** |
| Partner OEMs and encourage them to support new hardware and rapidly produce drivers etc. |
| Identify existing early adopters and capture their learning. How did they do it successfully? What are the critical success factors? |
| Target large organisations - ICT and other - and raise their awareness of FLOSS. Showcase the success stories. Reduce FUD, involve them in conferences and workshops to explore synergies. |
| Raise the debate, extend and update the sections on Business Models and IPR in this document. |
| Migration guidelines. Strategy development guides. Capture and communicate best practices. |
| Support SMMEs in accessing and leveraging the benefits of FLOSS. |
| Regulate selling of PCs with only MS Windows installed. Ensure buyers have the option not to buy a proprietary operating system. |
| |
| **Sectorial: agriculture, environment, water, etc.** |
| Target sectors such as Environment and Agriculture – workshop mutual needs, raise awareness of FLOSS issues, benefits, opportunities and existing relevant applications - e.g. open standards based open source Geographical Information Systems and associated tools. |
| Health: prevent deaths such as infant mortality via information access and sharing of resources. Install open source and standards based health management systems. |

| |
|---|
| **Education** |
| A National Strategy on ICT and education is needed - from general literacy to ICT literacy. Update the paradigm and promote supportive tools such as Moodle supporting a social constructivist approach etc. FLOSS in all aspects of education. Update content, empower communities to develop their own content. Massivisation of education - reaching all. Creative Commons licensing. |
| Analysis of the implications of proprietary software supplied free of charge to schools. |
| Promote resource sharing among schools, community centres, etc. (infrastructure, systems, support, knowledge, …). E.g. extend the clustering model employed by Shoolnet Namibia and the work of TSF. |

# X Conclusions

The rich variety of collaborative processes which have evolved to create open standards, open content and open source software are establishing new paradigms for all aspects of social development and economic growth. If these are not better recognised and understood by a much wider range of stakeholders, there is a severe risk that our currently unbalanced approach to IPR will stifle the organic, sustainable development and growth that we all are seeking. Indeed, the challenge of the complexity of the multiple feedback relationships between innovation, global goods, intellectual property, sustainable development, economic growth and quality of life needs to be embraced with full commitment. We cannot afford to over simplify these issues any longer.

Large private sector organisations should, at the very least, be objectively considering the benefits they would experience and extend to society if they were to migrate to FLOSS. SMMEs should take advantage of FLOSS to minimise the cost barriers to their use of ICTs, thereby benefiting not only themselves and their customers, but also the national economy via the multiplier effect.

An even more attractive vision is the central role FLOSS might play in establishing a vibrant, relevant and growing indigenous ICT industry in South Africa, particularly via small, local companies (individuals or start-ups). However, we must choose to provide appropriate FLOSS training and skills (and related resources) for the most disadvantaged communities via strategic and innovative methods at the national level.

Indeed, FLOSS should be seen as an essential component of all ICT for development initiatives, particularly for the sustainable benefit of the poor. But to achieve this, South Africa and like-minded influential developing countries should work with global institutions (many of which currently favour proprietary ICTs) in the pro-active establishment of de-facto, open, ICT for development standards.

The promise is that FLOSS is destined to have an equally revolutionary impact over the next 10 years as the Internet has had over the past 10 years. Although the developing world, including South Africa, was relegated to a following, copying and catch-up role during the internet revolution, we can become major players in the emerging FLOSS revolution. But we must choose to do so across all corners of society, and now.

# APPENDIX ONE: FLOSS Definitions

## *Open Source* (Di Bona et al. 1999, p. 253)

Open source doesn't just mean access to the source code. The distribution terms of an open-source program must comply with the following criteria:

1. Free Redistribution
The license may not restrict any party from selling or giving away the software as a component of an aggregate software distribution containing programs from several different sources. The license may not require a royalty or other fee for such sale.

2. Source Code
The program must include source code, and must allow distribution in source code as well as compiled form. Where some form of a product is not distributed with source code, there must be a well-publicized means of downloading the source code, without charge, via the Internet. The source code must be the preferred form in which a programmer would modify the program. Deliberately obfuscated source code is not allowed. Intermediate forms such as the output of a pre-processor or translator are not allowed.

3. Derived Works
The license must allow modifications and derived works, and must allow them to be distributed under the same terms as the license of the original software.

4. Integrity of the Author's Source Code
The license may restrict source code from being distributed in modified form only if the license allows the distribution of "patch files" with the source code for the purpose of modifying the program at build time. The license must explicitly permit distribution of software built from modified source code. The license may require derived works to carry a different name or version number from the original software.

5. No Discrimination Against Persons or Groups
The license must not discriminate against any person or group of persons.

6. No Discrimination Against Fields of Endeavour
The license must not restrict anyone from making use of the program in a specific field of endeavour. For example, it may not restrict the program from being used in a business, or from being used for genetic research.

7. Distribution of License
The rights attached to the program must apply to all to whom the program is redistributed without the need for execution of an additional license by those parties.

8. License Must Not Be Specific to a Product
The rights attached to the program must not depend on the program's being part of a particular software distribution. If the program is extracted from that distribution and used or distributed within the terms of the program's license, all parties to whom the program is redistributed should have the same rights as those that are granted in conjunction with the original software distribution.

9. License Must Not Contaminate Other Software
The license must not place restrictions on other software that is distributed along with the licensed software. For example, the license must not insist that all other programs distributed on the same medium must be open-source software.

## *Free Software*

Free software is a matter of the users' freedom to run, copy, distribute, study, change and improve the software. More precisely, it refers to four kinds of freedom, for the users of the software:

- The freedom to run the program, for any purpose (freedom 0).
- The freedom to study how the program works, and adapt it to your needs (freedom 1). Access to the source code is a precondition for this.
- The freedom to redistribute copies so you can help your neighbor (freedom 2).
- The freedom to improve the program, and release your improvements to the public, so that the whole community benefits (freedom 3). Access to the source code is a precondition for this.

[http://www.gnu.org/philosophy/free-sw.html]

# APPENDIX TWO: Letter from Juan Alberto González, G.M. Microsoft Peru[45]

San Isidro, March 21st 2002
To Mr. Edgar Villanueva Nuñez - Congressman of the Republic of Peru

Dear sir:

First of all, we want to thank you for the chance you gave us to inform you about our work in this country for benefit of the public sector, always looking for the best ways to implement programs that will let us consolidate the initiatives of modernization and transparency in the State.

In fact, thanks to our meeting today you are aware of our global achievements at the international level in the design of new services for the citizen, within the framework of a model State that respects and protects intellectual property.

The actions we talked about are part of a global initiative, and today there exist several experiences which have let us collaborate with programs supporting the State and community in the adoption of technology as a strategic element impacting the quality of life of the citizens.

Furthermore, as arranged in this meeting, we assisted the forum organized in the Congress on March 6th regarding the law project that you are leading, wherein we got the chance to listen to several presentations which lead us now to explain our position so you have a wider grasp of the real situation.

The bill makes it compulsory for all public bodies to use only free software, that is to say open source software, which breaches the principles of equality before the law, that of non-discrimination and the right of free private enterprise, freedom of industry and of contract, protected by the constitution.

The bill, by making the use of open source software compulsory, would establish discriminatory and non competitive practices in the contracting and purchasing by public bodies, violating the base principles of the "Law of State Contracting and Acquisitions" (Number 26850)

So, by compelling the State to favour a business model based entirely on open source, the bill would only discourage the local and international manufacturing companies, which are the ones which really undertake important expenditures, create a significant number of direct and indirect jobs, as well as contributing to the GNP, as opposed to a model of open source software which tends to have an ever weaker economic impact, since it mainly creates jobs in the service sector.

The bill imposes the use of open source software without considering the dangers that this can bring from the point of view of security, guarantee, and possible violation of the intellectual property rights of third parties.

The bill uses the concept of open source software incorrectly, since it does not necessarily imply that the software is free or of zero cost, and so arrives at mistaken conclusions regarding State savings, with no cost-benefit analysis to validate its position.

It is wrong to think that Open source software is free of charge. Research by the Gartner Group (an important investigator of the technological market recognized at world level) has shown that the cost of purchase of software (operating system and applications) is only 8% of the total cost which firms and institutions take on for a rational and truly beneficial use of the technology. The other 92% consists of: installation costs, enabling, support, maintenance, administration, and down-time.

The bill imposes the use of open source software without considering the dangers that this can bring from the point of view of security, guarantee, and possible violation of the intellectual property rights of third parties.

One of the arguments behind the bill is the supposed freedom from costs of open-source software, compared with the costs of commercial software, without taking into account the fact that there exist types of volume licensing which can be highly advantageous for the State, as has happened in other countries.

In addition, the alternative adopted by the bill (i) is clearly more expensive, due to the high costs of software migration, and (ii) puts at risk compatibility and interoperability of the IT platforms within the State, and between the State and the private sector, given the hundreds of versions of open source software on the market.

The majority of open source code does not offer adequate levels of service nor the guarantee from recognized manufacturers of high productivity on the part of the users, which has led various public organizations to retract their decision to go with an open source software solution and to use commercial software in its place.

The bill demotivates the creativity of the Peruvian software industry, which invoices 40 million US$/year, exports 4 million US$ (10th in ranking among non-traditional exports, more than handicrafts) and is a source of highly qualified employment. With a law that incentivates the use of open source, software programmers lose their intellectual property rights and their main source of payment.

[45] This English translation of the letter to Congressman Nuñez is taken from www.opensource.org where the original letter in Spanish is available http://www.opensource.org/docs/msFUD_to_peru.php.

Open source software, since it can be distributed without charge, does not allow the generation of income for its developers through exports. In this way, the multiplier effect of the sale of software to other countries is weakened, and so in turn is the growth of the industry, while Government rules ought on the contrary to stimulate local industry.

In the Forum, the use of open source software in education was discussed, without mentioning the complete collapse of this initiative in a country like Mexico, where precisely the State employees who founded the project now state that open source software did not make it possible to offer a learning experience to pupils in the schools, did not take into account the capability at a national level to give adequate support to the platform, and that the software did not and does not allow for the levels of platform integration that now exist in schools.

If open source software satisfies all the requirements of State bodies, why do you need a law to adopt it? Shouldn't it be the market which decides freely which products give most benefits or value?

I really want to thank you for your attention to this letter, and we want to reiterate our interest in meeting you to explain to you in more detail our point of view about the bill you have presented, and to be at your complete disposal to share experiences and information which we are sure can help better analyse and implement an initiative looking to modernization and transparency of the State for the benefit of the citizen.

Sincerely,
Juan Alberto González
General Manager
Microsoft Perú

# APPENDIX THREE: Reply to Mr. González Letter from Congressman NUÑEZ[46]

Lima, 8th of April, 2002
To: Señor JUAN ALBERTO GONZÁLEZ - General Manager of Microsoft Perú

Dear Sir:

First of all, I thank you for your letter of March 25, 2002 in which you state the official position of Microsoft relative to Bill Number 1609, Free Software in Public Administration, which is indubitably inspired by the desire for Peru to find a suitable place in the global technological context. In the same spirit, and convinced that we will find the best solutions through an exchange of clear and open ideas, I will take this opportunity to reply to the commentaries included in your letter.

While acknowledging that opinions such as yours constitute a significant contribution, it would have been even more worthwhile for me if, rather than formulating objections of a general nature (which we will analyze in detail later) you had gathered solid arguments for the advantages that proprietary software could bring to the Peruvian State, and to its citizens in general, since this would have allowed a more enlightening exchange in respect of each of our positions.

With the aim of creating an orderly debate, we will assume that what you call "open source software" is what the Bill defines as "free software", since there exists software for which the source code is distributed together with the program, but which does not fall within the definition established by the Bill; and that what you call "commercial software" is what the Bill defines as "proprietary" or "unfree", given that there exists free software which is sold in the market for a price like any other good or service.

It is also necessary to make it clear that the aim of the Bill we are discussing is not directly related to the amount of direct savings that can by made by using free software in state institutions. That is in any case a marginal aggregate value, but in no way is it the chief focus of the Bill. The basic principles which inspire the Bill are linked to the basic guarantees of a state of law, such as:

Free access to public information by the citizen.
Permanence of public data.
Security of the State and citizens.

To guarantee the free access of citizens to public information, it is indispensable that the encoding of data is not tied to a single provider. The use of standard and open formats gives a guarantee of this free access, if necessary through the creation of compatible free software.

To guarantee the permanence of public data, it is necessary that the usability and maintenance of the software does not depend on the goodwill of the suppliers, or on the monopoly conditions imposed by them. For this reason the State needs systems the development of which can be guaranteed due to the availability of the source code.

To guarantee national security or the security of the State, it is indispensable to be able to rely on systems without elements which allow control from a distance or the undesired transmission of information to third parties. Systems with source code freely accessible to the public are required to allow their inspection by the State itself, by the citizens, and by a large number of independent experts throughout the world. Our proposal brings further security, since the knowledge of the source code will eliminate the growing number of programs with *spy code*.

In the same way, our proposal strengthens the security of the citizens, both in their role as legitimate owners of information managed by the state, and in their role as consumers. In this second case, by allowing the growth of a widespread availability of free software not containing *spy code* able to put at risk privacy and individual freedoms.

In this sense, the Bill is limited to establishing the conditions under which the state bodies will obtain software in the future, that is, in a way compatible with these basic principles.

From reading the Bill it will be clear that once passed:

- the law does not forbid the production of proprietary software
- the law does not forbid the sale of proprietary software
- the law does not specify which concrete software to use
- the law does not dictate the supplier from whom software will be bought
- the law does not limit the terms under which a software product can be licensed.

What the Bill does express clearly, is that, for software to be acceptable for the state it is not enough that it is technically capable of fulfilling a task, but that further the contractual conditions must satisfy a series of requirements regarding the license, without which the State cannot guarantee the citizen adequate processing of his data, watching over its integrity, confidentiality, and accessibility throughout time, as these are very critical aspects for its normal functioning.

We agree, Mr. Gonzalez, that information and communication technology have a significant impact on the quality of life of the citizens (whether it be positive or negative). We surely also agree that the basic values I have pointed out above are fundamental in

[46] This English translation of Congressman Nuñez reply is taken from www.opensource.org where the original letter in Spanish is available http://www.opensource.org/docs/ peru_and_ms.php

a democratic state like Peru. So we are very interested to know of any other way of guaranteeing these principles, other than through the use of free software in the terms defined by the Bill.

As for the observations you have made, we will now go on to analyze them in detail:

Firstly, you point out that: "1. The bill makes it compulsory for all public bodies to use only free software, that is to say open source software, which breaches the principles of equality before the law, that of non-discrimination and the right of free private enterprise, freedom of industry and of contract, protected by the constitution."

This understanding is in error. The Bill in no way affects the rights you list; it limits itself entirely to establishing conditions for the use of software on the part of state institutions, without in any way meddling in private sector transactions. It is a well established principle that the State does not enjoy the wide spectrum of contractual freedom of the private sector, as it is limited in its actions precisely by the requirement for transparency of public acts; and in this sense, the preservation of the greater common interest must prevail when legislating on the matter.

The Bill protects equality under the law, since no natural or legal person is excluded from the right of offering these goods to the State under the conditions defined in the Bill and without more limitations than those established by the Law of State Contracts and Purchasing (T.U.O. by Supreme Decree No. 012-2001-PCM).

The Bill does not introduce any discrimination whatever, since it only establishes *how* the goods have to be provided (which is a state power) and not *who* has to provide them (which would effectively be discriminatory, if restrictions based on national origin, race religion, ideology, sexual preference etc. were imposed). On the contrary, the Bill is decidedly antidiscriminatory. This is so because by defining with no room for doubt the conditions for the provision of software, it prevents state bodies from using software which has a license including discriminatory conditions.

It should be obvious from the preceding two paragraphs that the Bill does not harm free private enterprise, since the latter can always choose under what conditions it will produce software; some of these will be acceptable to the State, and others will not be since they contradict the guarantee of the basic principles listed above. This free initiative is of course compatible with the freedom of industry and freedom of contract (in the limited form in which the State can exercise the latter). Any private subject can produce software under the conditions which the State requires, or can refrain from doing so. Nobody is forced to adopt a model of production, but if they wish to provide software to the State, they must provide the mechanisms which guarantee the basic principles, and which are those described in the Bill.

By way of an example: nothing in the text of the Bill would prevent your company offering the State bodies an office "suite", under the conditions defined in the Bill and setting the price that you consider satisfactory. If you did not, it would not be due to restrictions imposed by the law, but to business decisions relative to the method of commercializing your products, decisions with which the State is not involved.

To continue; you note that:" 2. The bill, by making the use of open source software compulsory, would establish discriminatory and non competitive practices in the contracting and purchasing by public bodies..."

This statement is just a reiteration of the previous one, and so the response can be found above. However, let us concern ourselves for a moment with your comment regarding "non-competitive ... practices."

Of course, in defining any kind of purchase, the buyer sets conditions which relate to the proposed use of the good or service. From the start, this excludes certain manufacturers from the possibility of competing, but does not exclude them "a priori", but rather based on a series of principles determined by the autonomous will of the purchaser, and so the process takes place in conformance with the law. And in the Bill it is established that *no one* is excluded from competing as far as he guarantees the fulfillment of the basic principles.

Furthermore, the Bill *stimulates* competition, since it tends to generate a supply of software with better conditions of usability, and to better existing work, in a model of continuous improvement.

On the other hand, the central aspect of competivity is the chance to provide better choices to the consumer. Now, it is impossible to ignore the fact that marketing does not play a neutral role when the product is offered on the market (since accepting the opposite would lead one to suppose that firms' expenses in marketing lack any sense), and that therefore a significant expense under this heading can influence the decisions of the purchaser. This influence of marketing is in large measure reduced by the bill that we are backing, since the choice within the framework proposed is based on the *technical merits* of the product and not on the effort put into commercialization by the producer; in this sense, competitiveness is increased, since the smallest software producer can compete on equal terms with the most powerful corporations.

It is necessary to stress that there is no position more anti-competitive than that of the big software producers, which frequently abuse their dominant position, since in innumerable cases they propose as a solution to problems raised by users: "update your software to the new version" (at the user's expense, naturally); furthermore, it is common to find arbitrary cessation of technical help for products, which, in the provider's judgment alone, are "old"; and so, to receive any kind of technical assistance, the user finds himself forced to migrate to new versions (with non-trivial costs, especially as changes in hardware platform are often involved). And as the whole infrastructure is based on proprietary data formats, the user stays "trapped" in the need to continue using products from the same supplier, or to make the huge effort to change to another environment (probably also proprietary).

You add: "3. So, by compelling the State to favor a business model based entirely on open source, the bill would only discourage the local and international manufacturing companies, which are the ones which really undertake important expenditures, create a

significant number of direct and indirect jobs, as well as contributing to the GNP, as opposed to a model of open source software which tends to have an ever weaker economic impact, since it mainly creates jobs in the service sector."

I do not agree with your statement. Partly because of what you yourself point out in paragraph 6 of your letter, regarding the relative weight of services in the context of software use. This contradiction alone would invalidate your position. The service model, adopted by a large number of companies in the software industry, is much larger in economic terms, and with a tendency to increase, than the licensing of programs.

On the other hand, the private sector of the economy has the widest possible freedom to choose the economic model which best suits its interests, even if this freedom of choice is often obscured subliminally by the disproportionate expenditure on marketing by the producers of proprietary software.

In addition, a reading of your opinion would lead to the conclusion that the State market is crucial and essential for the proprietary software industry, to such a point that the choice made by the State in this bill would completely eliminate the market for these firms. If that is true, we can deduce that the State must be subsidizing the proprietary software industry. In the unlikely event that this were true, the State would have the right to apply the subsidies in the area it considered of greatest social value; it is undeniable, in this improbable hypothesis, that if the State decided to subsidize software, it would have to do so choosing the free over the proprietary, considering its social effect and the rational use of taxpayers money.

In respect of the jobs generated by proprietary software in countries like ours, these mainly concern technical tasks of little aggregate value; at the local level, the technicians who provide support for proprietary software produced by transnational companies do not have the possibility of fixing bugs, not necessarily for lack of technical capability or of talent, but because they do not have access to the source code to fix it. With free software one creates more technically qualified employment and a framework of free competence where success is only tied to the ability to offer good technical support and quality of service, one stimulates the market, and one increases the shared fund of knowledge, opening up alternatives to generate services of greater total value and a higher quality level, to the benefit of all involved: producers, service organizations, and consumers.

It is a common phenomenon in developing countries that local software industries obtain the majority of their takings in the service sector, or in the creation of "ad hoc" software. Therefore, any negative impact that the application of the Bill might have in this sector will be more than compensated by a growth in demand for services (as long as these are carried out to high quality standards). If the transnational software companies decide not to compete under these new rules of the game, it is likely that they will undergo some decrease in takings in terms of payment for licenses; however, considering that these firms continue to allege that much of the software used by the State has been illegally copied, one can see that the impact will not be very serious. Certainly, in any case their fortune will be determined by market laws, changes in which cannot be avoided; many firms traditionally associated with proprietary software have already set out on the road (supported by copious expense) of providing services associated with free software, which shows that the models are not mutually exclusive.

With this bill the State is deciding that it needs to preserve certain fundamental values. And it is deciding this based on its sovereign power, without affecting any of the constitutional guarantees. If these values could be guaranteed without having to choose a particular economic model, the effects of the law would be even more beneficial. In any case, it should be clear that the State does not choose an economic model; if it happens that there only exists one economic model capable of providing software which provides the basic guarantee of these principles, this is because of historical circumstances, not because of an arbitrary choice of a given model.

Your letter continues: "4. The bill imposes the use of open source software without considering the dangers that this can bring from the point of view of security, guarantee, and possible violation of the intellectual property rights of third parties."

Alluding in an abstract way to "the dangers this can bring", without specifically mentioning a single one of these supposed dangers, shows at the least some lack of knowledge of the topic. So, allow me to enlighten you on these points.

On security:
National security has already been mentioned in general terms in the initial discussion of the basic principles of the bill. In more specific terms, relative to the security of the software itself, it is well known that all software (whether proprietary or free) contains errors or "bugs" (in programmers' slang). But it is also well known that the bugs in free software are fewer, and are fixed much more quickly, than in proprietary software. It is not in vain that numerous public bodies responsible for the IT security of state systems in developed countries require the use of free software for the same conditions of security and efficiency.

What is impossible to prove is that proprietary software is more secure than free, without the public and open inspection of the scientific community and users in general. This demonstration is impossible because the model of proprietary software itself prevents this analysis, so that any guarantee of security is based only on promises of good intentions (biased, by any reckoning) made by the producer itself, or its contractors.

It should be remembered that in many cases, the licensing conditions include Non-Disclosure clauses which prevent the user from publicly revealing security flaws found in the licensed proprietary product.

In respect of the guarantee:
As you know perfectly well, or could find out by reading the "End User License Agreement" of the products you license, in the great majority of cases the guarantees are limited to replacement of the storage medium in case of defects, but in no case is compensation given for direct or indirect damages, IFLOSS of profits, etc... If as a result of a security bug in one of your products, not fixed in time by yourselves, an attacker managed to compromise crucial State systems, what guarantees, reparations and compensation would your company make in accordance with your licensing conditions? The guarantees of proprietary software,

inasmuch as programs are delivered ``AS IS'', that is, in the state in which they are, with no additional responsibility of the provider in respect of function, in no way differ from those normal with free software.

On Intellectual Property:
Questions of intellectual property fall outside the scope of this bill, since they are covered by specific other laws. The model of free software in no way implies ignorance of these laws, and in fact the great majority of free software is covered by copyright. In reality, the inclusion of this question in your observations shows your confusion in respect of the legal framework in which free software is developed. The inclusion of the intellectual property of others in works claimed as one's own is not a practice that has been noted in the free software community; whereas, unfortunately, it has been in the area of proprietary software. As an example, the condemnation by the Commercial Court of Nanterre, France, on 27th September 2001 of Microsoft Corp. to a penalty of 3 million francs in damages and interest, for violation of intellectual property (piracy, to use the unfortunate term that your firm commonly uses in its publicity).

You go on to say that: "The bill uses the concept of open source software incorrectly, since it does not necessarily imply that the software is free or of zero cost, and so arrives at mistaken conclusions regarding State savings, with no cost-benefit analysis to validate its position."

This observation is wrong; in principle, freedom and lack of cost are orthogonal concepts: there is software which is proprietary and charged for (for example, MS Office), software which is proprietary and free of charge (MS Internet Explorer), software which is free and charged for (Red Hat, SuSE etc GNU/Linux distributions), software which is free and not charged for (Apache, Open Office, Mozilla), and even software which can be licensed in a range of combinations (MySQL).

Certainly free software is not necessarily free of charge. And the text of the bill does not state that it has to be so, as you will have noted after reading it. The definitions included in the Bill state clearly *what* should be considered free software, at no point referring to freedom from charges. Although the possibility of savings in payments for proprietary software licenses are mentioned, the foundations of the bill clearly refer to the fundamental guarantees to be preserved and to the stimulus to local technological development. Given that a democratic State must support these principles, it has no other choice than to use software with publicly available source code, and to exchange information only in standard formats.

If the State does not use software with these characteristics, it will be weakening basic republican principles. Luckily, free software also implies lower total costs; however, even given the hypothesis (easily disproved) that it was more expensive than proprietary software, the simple existence of an effective free software tool for a particular IT function would oblige the State to use it; not by command of this Bill, but because of the basic principles we enumerated at the start, and which arise from the very essence of the lawful democratic State.

You continue: "6. It is wrong to think that Open source software is free of charge. Research by the Gartner Group (an important investigator of the technological market recognized at world level) has shown that the cost of purchase of software (operating system and applications) is only 8% of the total cost which firms and institutions take on for a rational and truly beneficial use of the technology. The other 92% consists of: installation costs, enabling, support, maintenance, administration, and down-time."

This argument repeats that already given in paragraph 5 and partly contradicts paragraph 3. For the sake of brevity we refer to the comments on those paragraphs. However, allow me to point out that your conclusion is logically false: even if according to Gartner Group the cost of software is on average only 8% of the total cost of use, this does not in any way deny the existence of software which is free of charge, that is, with a licensing cost of zero.

In addition, in this paragraph you correctly point out that the service components and losses due to down-time make up the largest part of the total cost of software use, which, as you will note, contradicts your statement regarding the small value of services suggested in paragraph 3. Now the use of free software contributes significantly to reduce the remaining life-cycle costs. This reduction in the costs of installation, support etc. can be noted in several areas: in the first place, the competitive service model of free software, support and maintenance for which can be freely contracted out to a range of suppliers competing on the grounds of quality and low cost. This is true for installation, enabling, and support, and in large part for maintenance. In the second place, due to the reproductive characteristics of the model, maintenance carried out for an application is easily replicable, without incurring large costs (that is, without paying more than once for the same thing) since modifications, if one wishes, can be incorporated in the common fund of knowledge. Thirdly, the huge costs caused by non-functioning software ("blue screens of death", malicious code such as virus, worms, and trojans, exceptions, general protection faults and other well-known problems) are reduced considerably by using more stable software; and it is well known that one of the most notable virtues of free software is its stability.

You further state that: "7. One of the arguments behind the bill is the supposed freedom from costs of open-source software, compared with the costs of commercial software, without taking into account the fact that there exist types of volume licensing which can be highly advantageous for the State, as has happened in other countries."

I have already pointed out that what is in question is not the cost of the software but the principles of freedom of information, accessibility, and security. These arguments have been covered extensively in the preceding paragraphs to which I would refer you.
On the other hand, there certainly exist types of volume licensing (although unfortunately proprietary software does not satisfy the basic principles). But as you correctly pointed out in the immediately preceding paragraph of your letter, they only manage to reduce the impact of a component which makes up no more than 8% of the total.

You continue: "8. In addition, the alternative adopted by the bill (I) is clearly more expensive, due to the high costs of software migration, and (II) puts at risk compatibility and interoperability of the IT platforms within the State, and between the State and the private sector, given the hundreds of versions of open source software on the market."
Let us analyze your statement in two parts. Your first argument, that migration implies high costs, is in reality an argument in favor of the Bill. Because the more time goes by, the more difficult migration to another technology will become; and at the same time, the

security risks associated with proprietary software will continue to increase. In this way, the use of proprietary systems and formats will make the State ever more dependent on specific suppliers. Once a policy of using free software has been established (which certainly, does imply some cost) then on the contrary migration from one system to another becomes very simple, since all data is stored in open formats. On the other hand, migration to an open software context implies no more costs than migration between two different proprietary software contexts, which invalidates your argument completely.

The second argument refers to "problems in interoperability of the IT platforms within the State, and between the State and the private sector" This statement implies a certain lack of knowledge of the way in which free software is built, which does not maximize the dependence of the user on a particular platform, as normally happens in the realm of proprietary software. Even when there are multiple free software distributions, and numerous programs which can be used for the same function, interoperability is guaranteed as much by the use of standard formats, as required by the bill, as by the possibility of creating interoperable software given the availability of the source code.

You then say that: "9. The majority of open source code does not offer adequate levels of service nor the guarantee from recognized manufacturers of high productivity on the part of the users, which has led various public organizations to retract their decision to go with an open source software solution and to use commercial software in its place."

This observation is without foundation. In respect of the guarantee, your argument was rebutted in the response to paragraph 4. In respect of support services, it is possible to use free software without them (just as also happens with proprietary software), but anyone who does need them can obtain support separately, whether from local firms or from international corporations, again just as in the case of proprietary software.

On the other hand, it would contribute greatly to our analysis if you could inform us about free software projects *established* in public bodies which have already been abandoned in favor of proprietary software. We know of a good number of cases where the opposite has taken place, but not know of any where what you describe has taken place.

You continue by observing that: "10. The bill discourages the creativity of the Peruvian software industry, which invoices 40 million US$/year, exports 4 million US$ (10th in ranking among non-traditional exports, more than handicrafts) and is a source of highly qualified employment. With a law that encourages the use of open source, software programmers lose their intellectual property rights and their main source of payment."

It is clear enough that nobody is forced to commercialize their code as free software. The only thing to take into account is that if it is not free software, it cannot be sold to the public sector. This is not in any case the main market for the national software industry. We covered some questions referring to the influence of the Bill on the generation of employment which would be both highly technically qualified and in better conditions for competition above, so it seems unnecessary to insist on this point.

What follows in your statement is incorrect. On the one hand, no author of free software loses his intellectual property rights, unless he expressly wishes to place his work in the public domain. The free software movement has always been very respectful of intellectual property, and has generated widespread public recognition of its authors. Names like those of Richard Stallman, Linus Torvalds, Guido van Rossum, Larry Wall, Miguel de Icaza, Andrew Tridgell, Theo de Raadt, Andrea Arcangeli, Bruce Perens, Darren Reed, Alan Cox, Eric Raymond, and many others, are recognized world-wide for their contributions to the development of software that is used today by millions of people throughout the world. On the other hand, to say that the rewards for authors rights make up the main source of payment of Peruvian programmers is in any case a guess, in particular since there is no proof to this effect, nor a demonstration of how the use of free software by the State would influence these payments.

You go on to say that: "11. Open source software, since it can be distributed without charge, does not allow the generation of income for its developers through exports. In this way, the multiplier effect of the sale of software to other countries is weakened, and so in turn is the growth of the industry, while Government rules ought on the contrary to stimulate local industry."

This statement shows once again complete ignorance of the mechanisms of and market for free software. It tries to claim that the market of sale of non- exclusive rights for use (sale of licenses) is the only possible one for the software industry, when you yourself pointed out several paragraphs above that it is not even the most important one. The incentives that the bill offers for the growth of a supply of better qualified professionals, together with the increase in experience that working on a large scale with free software within the State will bring for Peruvian technicians, will place them in a highly competitive position to offer their services abroad.

You then state that: "12. In the Forum, the use of open source software in education was discussed, without mentioning the complete collapse of this initiative in a country like Mexico, where precisely the State employees who founded the project now state that open source software did not make it possible to offer a learning experience to pupils in the schools, did not take into account the capability at a national level to give adequate support to the platform, and that the software did not and does not allow for the levels of platform integration that now exist in schools."

In fact Mexico has gone into reverse with the Red Escolar (Schools Network) project. This is due precisely to the fact that the driving forces behind the Mexican project used license costs as their main argument, instead of the other reasons specified in our project, which are far more essential. Because of this conceptual mistake, and as a result of the lack of effective support from the SEP (Secretary of State for Public Education), the assumption was made that to implant free software in schools it would be enough to drop their software budget and send them a CD ROM with Gnu/Linux instead. Of course this failed, and it couldn't have been otherwise, just as school laboratories fail when they use proprietary software and have no budget for implementation and maintenance. That's exactly why our bill is not limited to making the use of free software mandatory, but recognizes the need to create a viable migration plan, in which the State undertakes the technical transition in an orderly way in order to then enjoy the advantages of free software.

You end with a rhetorical question: "13. If open source software satisfies all the requirements of State bodies, why do you need a law to adopt it? Shouldn't it be the market which decides freely which products give most benefits or value?"

We agree that in the private sector of the economy, it must be the market that decides which products to use, and no state interference is permissible there. However, in the case of the public sector, the reasoning is not the same: as we have already established, the state archives, handles, and transmits information which does not belong to it, but which is entrusted to it by citizens, who have no alternative under the rule of law. As a counterpart to this legal requirement, the State must take extreme measures to safeguard the integrity, confidentiality, and accessibility of this information. The use of proprietary software raises serious doubts as to whether these requirements can be fulfilled, lacks conclusive evidence in this respect, and so is not suitable for use in the public sector.

The need for a law is based, firstly, on the realization of the fundamental principles listed above in the specific area of software; secondly, on the fact that the State is not an ideal homogeneous entity, but made up of multiple bodies with varying degrees of autonomy in decision making. Given that it is inappropriate to use proprietary software, the fact of establishing these rules in law will prevent the personal discretion of any state employee from putting at risk the information which belongs to citizens. And above all, because it constitutes an up-to-date reaffirmation in relation to the means of management and communication of information used today, it is based on the republican principle of openness to the public.

In conformance with this universally accepted principle, the citizen has the right to know all information held by the State and not covered by well- founded declarations of secrecy based on law. Now, software deals with information and is itself information. Information in a special form, capable of being interpreted by a machine in order to execute actions, but crucial information all the same because the citizen has a legitimate right to know, for example, how his vote is computed or his taxes calculated. And for that he must have free access to the source code and be able to prove to his satisfaction the programs used for electoral computations or calculation of his taxes.

I wish you the greatest respect, and would like to repeat that my office will always be open for you to expound your point of view to whatever level of detail you consider suitable.

Cordially,
DR. EDGAR DAVID VILLANUEVA NUÑEZ
Congressman of the Republic of Perú.

## APPENDIX FOUR: Methods of cost and benefit analysis

In cases where it remains important to be able to evaluate the costs and benefits of specific potential FLOSS solutions, the following two methods of cost and benefit analysis are useful.

***Total Cost of Ownership (TCO):***
Total cost of ownership tries to capture all the real costs of a technology. There are two principal types of expenditures on software: direct and indirect. Direct costs include licensing fees, installation costs, training and support. Indirect costs are consequential costs of using the software, including hardware upgrades and costs associated with ensuring backward compatibility. TCO assesses both human and technological costs of the software across its life cycle. A shortcoming of TCO is its focus on the costs side of the equation. Because of this, methodologies, such as return on investment, which attempt to assess benefits gained are often employed.

***Return on Investment (ROI):***
ROI compares the net investment relative to the incremental benefits of that investment over some period. A challenge in applying ROI is in establishing causal relationships between an investment and its benefits. This means not just a comparison of technological costs is needed, but that a comparative analysis of organisational process and productivity improvements are required. ROI should therefore be seen as a conceptually simple but potentially analytically complex cost and benefit assessment tool which can ultimately be presumptive and even subjective. There are a host of ROI analyses whose comparative explanatory power depends on an organisation's relative strategic and temporal concerns[47].

[47] For further discussion of OSS cost and benefit methodologies see Blume *et al.* (2003).

## References and Resources


Blume, Roy, Natalie Bryden, Brian Neilson, and Mark Rotter (2003) "Designing and managing a framework for assessing results of use to FLOSS in South Africa: Phase 1", *BMI Tech Knowledge group Report*, No. BMI-T #SITA2003A.

Bonaccorsi, Andrea and Cristina Rossi (2003) "Why Open source software can succeed", *Research Policy*, 32, pp. 1243-1258.

Boyle, James (2002) "Fencing off ideas: enclosure and the disappearance of the public domain", Daedalus, Available at: http://james-boyle.com/daedalus.pdf

Cabinet Office (2001) "e-government interoperability framework - version 3", Available at: http://www.govtalk.gov.uk/interoperability/egif.asp

Castells, M & Himanen, P (2002). "The Information Society and the Welfare State: The Finnish Model", Oxford University Press.

Cowan, Robin (2001) "Expert systems: aspects of and limitations to the codifiability of knowledge", *Research Policy*, 30, pp.1355-1372.

Cowan, Robin and Elad Harrison (2001a) "Intellectual Property Rights in a Knowledge-Based Economy", *MERIT-Infonomics Research Memorandum*, No. 2001-027. Available at: http://www.merit.unimaas.nl/publications/index.php

Cowan, Robin and Elad Harrison (2001b) "Protecting The Digital Endeavour: Prospects For Intellectual Property Rights In The Information Society", *MERIT-Infonomics Research Memorandum*, No. 2001-028. Available at: http://www.merit.unimaas.nl/publications/index.php

David, Paul A. (2000) "A Tragedy of the Public Knowledge 'Commons'? Global Science, Intellectual Property and the Digital Technology Boomerang", *SIEPR Discussion Paper*, No. 00-02.

David, Paul A. (1993) "Intellectual Property Institutions and the Panda's Thumb: Patents, Copyrights and Trade Secrets in Economic Theory and History" *in Wallerstein et al. (eds.)* pp.19-63.

David, Paul A. and Dominique Foray (2001) "An Introduction to the Economy of the Knowledge Society", *MERIT-Infonomics Research Memorandum*, No. 2001-041. Available at: http://www.merit.unimaas.nl/publications/index.php

Department of Public Services and Administration (2002) "Handbook on Minimum Information Interoperability Standards (MIOS)". Available at: http://www.dpsa.gov.za/e-gov/2002docs/MIOS-Handbook16April'02.pdf

DiBona, Chris, Sam Ockman and Mark Stone (eds.) (1999) "Open Sources: Voices from

the Open Source Revolution", Sebastopol, California: O'Reilly.

DiBona, Chris, Sam Ockman and Mark Stone (1999) "Introduction" *in DiBona et al. (eds.)*, pp. 1-18.

Esselaar P. James, T. Miller, J. and Sibthorpe, G. (2002). "ICT Diffusion and Applications in Eight Industry Sectors in South Africa.". *TIPS Research Paper*, Available at: http://www. tips.org.za/research/dtipsp.

FLOSS (2003) "Free/Libre and Open source Software: Survey and Study: Final Report", Available at: http://FLOSS.infonomics.nl/ .

Ghosh, Rishab (1998) "Cooking pot markets: an economic model for the trade in free goods and services on the Internet", *First Monday*, 3(3), March 1998, Available on line at: http://www.firstmonday.org/issues/issue3_3/ghosh/index.html.

Gilsing, Victor and Marianne van der Steen (2003) "Universal Opitmality versus Local Optimality in the IPR-debate: The Balancing Act of Knowledge Production and Knowledge Diffusion in the Knowledge Economy", paper presented at the *EAEPE Conference 'The Information Society - Understanding its Institutions Interdisciplinary'*, 7-10 November, Maastricht.

Granstrand, Ove (2000) "The shift towards intellectual capitalism - the role of infocom technologies", *Research Policy*, 29, pp. 1061-1080.

Grossman, Wendy M. (2002) "Stolen code", *New Scientist*, 28 September, pp. 36-39.

Hahn, Robert W. (ed.) (2003) "Government Policy Towards Open Source Software". Washington D.C.: The Brookings Institution.

Heller, Michael A. and Rebecca S. Eisenberg (1998) "Can Patents Deter Innovation? The Anticommons in Biomedical Research" *Science*, 280(5364), pp. 698-701.

Himmanen, Pekka, Torvalds L. and Castells M (2001) "The Hacker Ethic and the Spirit of the Information Age". Random House.

International Labour Organisation (2001) "World Employment Report 2001: Life at work in the Information Economy". Geneva: ILO.

ISETT SETA (2002) "ICT skills audit finds shortage" Department of Labour, *ITWEB*, 29 August 2002. Available at: http://www.itweb.co.za/sections/business/2002/0208291159.asp?O=TE /.

Krisgnamurthy, Sandeep (2003) "A managerial overview of open source software", *Business Horizons*, September-October 2003, pp. 47-56.

Lakhani, Karim and Eric von Hippel (2000) "How Open source software works: 'Free'

user-to-user assistance", *MIT Sloan School of Management Working Paper*, No. 4117. Massachusetts Institute of Technology.

Lerner, Josh and Jean Tirole (2000) "The Simple Economics of Open Source", *NBER Working Paper*, No. 7600. National Bureau of Economic Research.

Lessig, Lawrence (2003) "Open Source Baselines: Compared to What?", in *Hahn (ed.*). Chapter Four, pp. 50-68.

Lohr, Steve (2003) "Web Group Backs Microsoft in Patent Suit", *New York Times: nytimes.com*, 29 October 2003. Available at: http://www.nytimes.com/

Malerba, Franco and Lugi Orsenigo (1997) "Technological Regimes and Sectoral Patterns of Innovative Activities" *Industrial and Corporate Change*, 6, pp. 83-117.

Moody, Glyn (2001) "Rebel Code: Linux and the Open Source Revolution". London: Allen Lane.

National Research Council (2003) "The Role of Scientific and Technical Data and Information in the Public Domain: Proceedings of a Symposium". Washington D.C.: National Academy Press.

National Research Council (2000) "The Digital Dilemma: Intellectual Property in the Information Age". Washington D.C.: National Academy Press.

National Research Council (1999) "A Question of Balance: Private Rights and the Public Interest in Scientific and Technical Databases". Washington D.C.: National Academy Press.

National Research Council (1997) "Bits of Power: Issues in Global Access to Scientific Data". Washington D.C.: National Academy Press.

Nuvolari, Alessandro (2003) "Open source software development: some historical perspectives", *ECIS Working Paper* No. 03.01. Available at: http://www.tm.tue.nl/ecis/

OSS Working Group (2003). "Using Open source software in the South African government", *Government IT Officers Council (GITOC)*, Version 3.3. Available at: http://www.oss.gov.za/docs/OSS_Strategy_v3.pdf

Peeling, Nic and Julian Stachell (2001) "Analysis of the Impact of Open Source Software", *UK Government Discussion Document*, Available at: http://www.govtalk.gov.uk/library

Perens, Bruce (1999) "The Open Source Definition", in *DiBona et al.(eds.)*, pp. 171-188.

Raymond, Eric S. (2003) "The Art of UNIX Programming". Boston: Addison-Wesley.

Raymond, Eric S. (2001) “The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary”. Sebastopol, California: O’Reilly.

Reichman, Jerome H. (1994) “Legal Hybrids Between the Patent and Copyright Paradigms”, *Colombia Law Review*, 94(8), pp. 2432-2558.

Royal Society (2003) “Keeping Science Open: the effects of intellectual property policy on the conduct of science”, *Royal Society Report*, April.

Spiegler, Israel (2003) “Technology and knowledge: bridging a ‘generating’ gap”, *Information & Management*, 40, pp. 533-539.

Stein, Pamela, Patel, Safiyya and Tshepe Doris (2003) “Cyberlaw for Civil Society: A Resource Guide”, edited by Pamela Stein; Published by: OSISA and Cheadle Thompson & Haysom Inc. Available at http://www.cth.co.za and http://www.osisa.org

Stephan, Paula E. (1996) “The Economics of Science” *Journal of Economic Literature*, 34, pp. 1199-1235.

NACI. (2002) “Open Software & Open Standards in South Africa – A Critical Issue for Addressing the Digital Divide”, Available at: http://www.naci.org.za/pdfs/opensource.pdf

Torvalds L and Diamond D (2001) “Just for Fun: The Story of an Accidental Revolutionary. HarperBusiness.

UNESCO (2002a) “EFA Global Monitoring Report 2002. Education for All. Is the World on Track?” Paris: UNESCO.

UNESCO (2002b) “Strategic Framework for Sub-Saharan Africa. In the Area of Teacher Training and Secondary Education Through Open and Distance Learning”. Paris: UNESCO.

Vance, Ashlee (2003) “HP offers Linux users SCO protection”, *The Register*, 24 September 2003. Available at: http://www.theregister.co.uk/content/61/33017.html.

Van der Merwe, D (1999) "The Dematerialization of Print and the Fate of Copyright“, p303, in *“International Review of Law, Computers & Technology",* Vol 13, No3 December 1999.

Wallerstein, Mitchel B. Roberta A. Schoen and Mary E. Mogee (eds.) “Global Dimensions of Intellectual Property Rights in Science and Technology”. Washington D.C.: National Academy Press.