Ernst-Abbe-Fachhochschule Jena
Hochschule für angewandte Wissenschaften
University of Applied Sciences

Research Papers

Workshop Proceedings
Sept. 15-17, 2012, Kabul/Afghanistan

Creating Awareness for the Use of OpenSource Systems in the Public Sector in Afghanistan

editors

Sayed Jahed Hussini
Balkh University, Mazar-e-Sharif/Afghanistan

Wolfgang F. Finke
Ernst-Abbe University of Applied Sciences, Jena/Germany

Journal 2/2013
Faculty of Business

NATO OTAN
This publication is supported by:
The NATO **Science** for **Peace** and **Security** Programme

Series: **Research Papers, Volume 2013, Journal 2**
Business Information Systems

Editors: Sayed Jahed Hussini
(Balkh University, Mazar-e-Sharif / Afghanistan)
Wolfgang F. Finke
(Ernst-Abbe University of Applied Sciences, Jena / Germany)

ISSN 1868-1697
ISBN 3-939046-32-9

Ernst-Abbe University of Applied Sciences
Carl-Zeiss-Promenade 2
07745 Jena / Germany
Phone: +49 3641 205 550
Fax: +49 3641 205 551

Place of publication: Jena / Germany

This publication is supported by the NATO Science for Peace and Security Programme.

Printed in Germany

Summary

The notion that Afghanistan at present is at a decisive point in time is well known. Information systems in Afghan public institutions are no exception. Mainly caused by the lack of available knowledge and competencies with regard to Open Source software, information systems infrastructures - by default - are build-up from scratch by using proprietary software. This strategy might not be sustainable in the medium or long term.

It was a lucky coincidence that the NATO Science for Peace and Security Programme was ready to provide the funding for a workshop about the utilization of Open Source systems in different application environments in Sept. 2012 in Kabul. We were able to assemble a group of about thirty Afghan and international IT/IS specialists (some of them representing the fledgling Afghan Open Source community) for a three days workshop.

Focal areas of presentations and discussions were Open Source systems in health care management, city administration, universities, justice, and additional topics in technical areas. There were discussions/presentations about regional cooperation in Central Asia and the status of Open Source projects/communities in Europe and Central Asia as well. We had lively exchanges of views and concepts and both parties – Afghan and European IT/IS specialists – enjoyed a host of new insights and ideas.

A mayor concern of the participants of the workshop and the training sessions was to find a way to systematically introduce and support Open Source systems in Afghan organizations. While there is already strong interest in Open Source systems in some of the larger public organizations in Afghanistan, professional support is available only for proprietary software.

Therefore, an important precondition for helping Afghan organizations to build-up and utilize Open Source systems in a professional way is the formation of an Afghan Center for Open Source Systems (ACOSS) in cooperation with Afghan and international stakeholders. Furthermore, such a competence center could accept important second level support functions for selected Open Source application systems.

Keywords

Open source systems, information systems, information management, open source software (OSS), IT infrastructure, computer systems, Afghan public institutions, ACOSS Afghan Center for Open Source Systems, WollMUX, health care management.

Preface

The notion that Afghanistan at present is at a decisive point in time is well known. Information systems in Afghan public institutions are no exception. Mainly caused by the lack of available knowledge and competencies with regard to Open Source software, information systems infrastructures - by default - are build-up from scratch by using proprietary software. This strategy might not be sustainable in the medium or long term.

There is right now an urgent need to utilize Open Source systems in Afghanistan and to build-up the necessary know-how – especially by educating Afghan IT-, IS-, and consulting specialists - to accomplish this goal. The introduction of professional information management structures to Afghan organizations is an important task as well: IT specialists usually lack the competencies to effectively manage organization-wide information systems or large-scale application development projects and Business Information Systems programs are not yet known at Afghan universities.

It was a lucky coincidence that the NATO Science for Peace and Security Programme was ready to provide the funding for a workshop about the utilization of Open Source systems in different application environments in Sept. 2012 in Kabul. We were able to assemble a group of about thirty Afghan and international IT/IS specialists (some of them representing the fledgling Afghan Open Source community) for a three days workshop.

Focal areas of presentations and discussions were Open Source systems in health care management, city administration, universities, justice, and additional topics in technical areas. There were discussions/presentations about regional cooperation in Central Asia and the status of Open Source projects/communities in Europe and Central Asia as well. We had lively exchanges of views and concepts and both parties – Afghan and European IT/IS specialists – enjoyed a host of new insights and ideas.

As a follow-up activity two training sessions (three days in Dec. 2012 and five days in Feb. 2013) were provided for experienced Afghan IT/IS administrators – with financial support of the NATO Science for Peace and Security Programme and with support of the University of Groningen/Netherlands, who allowed Dr. Frank Brokken to join the effort as a trainer. We were able to provide several up-to-date books about the administration of Open Source infrastructures to the training participants, too. This was an important precondition for self-guided and continuing learning of the training participants and for enabling them to set-up their first test-environments in Afghanistan. Dr. Frank Brokken

brought about 1.5 Terra-byte of OpenSource software (Debian/Ubuntu mirror servers) with him, so that we could provide a fairly sophisticated lab environment for our students.

A mayor concern of the participants of the workshop and the training sessions was to find a way to systematically introduce and support Open Source systems to Afghan organizations. While there is already strong interest in Open Source systems in some larger public organizations in Afghanistan, professional support is available only for proprietary software. Therefore, an important precondition for helping Afghan organizations to build-up and utilize Open Source systems in a professional way is the formation of an Afghan Center for Open Source Systems (ACOSS) in cooperation with Afghan and international stakeholders. Furthermore, such a competence center could accept important second level support functions for selected Open Source application systems (e.g., health care management systems which were proposed for Afghan hospitals by scientists from the University of Oslo/Norway).

Last-but-not-least, we want to thank the large number of people who provided essential support for getting the workshop and the training sessions up and running. Sometimes this support was really indispensable and vital in the real sense of the word. Our special thanks go to Dr. Saidmohammad Tingar of the University of Kabul, Mrs. Sandra Maria Wassong and Mr. Marcel Consten of the German Embassy in Kabul, Dr. Yahya Wardak, CIM Expert at Ministry of Higher Education in Kabul, and Dr. Walter Kaffenberger, NATO Science for Peace and Security Programme.

Table of Contents

1. Why does Afghanistan need Open Source systems?

Wolfgang F. Finke

Ernst-Abbe University of Applied Sciences, Jena/Germany

Summary

After decades of war and a focus on military activities during the last years there is a need to focus on the reconstruction of Afghan civil life and Afghanistan's public organizations. They need to be enabled to deliver efficient and effective services to the Afghan state, its citizens, and its private organizations. To a high degree tasks in public organizations are information processing and communication tasks. Therefore, like banks and insurance companies, many public institutions are information factories which collect and maintain large pools of data and knowledge and use it to support information work and communication processes and to generate/disseminate information.

Afghan public organizations without a well-developed information technology (IT) infrastructure and computer-supported information systems (IS) cannot be expected to reach acceptable levels of performance. They will be inefficient, intransparent, non-manageable, and will provide poor services to their “customers” - Afghan citizens and other public or private organizations.

In many parts of the world organizations struggle to convert their IT/IS infrastructures to Open Source systems. In some cases, large-scale government-sponsored projects are under way to utilize Open Source software. Due to weak

information management capacities and resource constraints things seem to go awfully wrong in Afghan public institutions during the present build-up phase of IT/IS infrastructures: Because of a general lack of knowledge and competencies, in most cases the investments in proprietary IT/IS infrastructures will lead to systems which will not be sustainable and will have to be rebuilt again from scratch in the years ahead (and with Open Source software at their core). Significant parts of IT/IS investments will be lost.

In most cases, Afghanistan's government organizations and public institutions will have no other chance but to rely on Open Source systems and infrastructures in the future. But utilizing Open Source systems will not solve every information systems problem in Afghan public organizations. There is a plethora of problems in the areas of professional IT/IS management, systematic development/implementation of work processes, education of end-users and IT/IS professionals, building up viable network infrastructures with sufficient bandwidth to connect geographically dispersed organizations and for communicating with the outside world, and creating/maintaining a viable database and computing infrastructure. These problems will have to be solved alongside the introduction of OSS to make IT/IS work well in Afghan organizations.

It is urgently required to build-up the necessary OSS expertise in Afghanistan and especially in Kabul. We propose ACOSS-PO - Afghan Center for Open Source Systems in Public Organizations which would be able to transfer the required Open Source knowledge and competencies to Afghanistan and could provide professional education/certification, consulting and IT/IS services to public organizations which want to set-up Linux- and Open Source-based information systems.

Keywords

Information factory, information systems, information management, open source software (OSS), IT infrastructure, office automation, work process design, computer systems, Afghan public institutions, ACOSS Afghan Center for Open Source Systems.

1.1 Afghan public organizations and their information systems

After decades of war and a focus on military activities during the last twelve years there is a need to focus on the reconstruction of Afghan civil life and Afghanistan's public organizations. They need to be enabled to deliver efficient and effective services to the Afghan state, its citizens, and its private organizations.

Government institutions and other public organizations (e.g. universities, hospitals, the Afghan army and police forces) are often relatively big and employ large numbers of people who communicate or collaborate within their organizations and beyond their boundaries to accomplish daily tasks (e.g. conducting the annual Kankor process, developing strategic plans, preparing security assessments by processing intelligence, planning and creating an Afghan higher education accreditation organization). The utilization of explicit and tacit knowledge is an important ingredient of these semi-structured and collaborative work processes. Besides collaboration tasks there are a number of high-volume and well-structured operational information tasks which constitute the core of an organization's work processes (e.g. patient management in a hospital, student and exam administration at a university, payroll processing for government employees at a ministry or for members of the Afghan National Army).

To a high degree tasks in public organizations are information processing and communication tasks – and not physical transformation or transportation tasks. Therefore, like banks and insurance companies, many public institutions are information factories which collect and maintain large pools of data and knowledge and use it to support information work and communication processes, and to generate/disseminate information. Efficient information factories require adequate management, functioning information work processes and information systems, capable information workers, and the proper “machinery” (IT infrastructure).

Afghan public organizations without a well-developed information technology (IT) infrastructure and computer-supported information systems (IS) cannot be expected to reach an acceptable level of performance. They will be inefficient, in-transparent, non-manageable, and will provide poor services to their “customers” - Afghan citizens and other public or private organizations. Furthermore,

an Afghan public institution can be seen as one of the cog-wheels in a larger gear box: If one of the Afghan public organizations (e.g. a ministry or a provincial administration) is not performing well, it is likely that this will impede the functioning of other public or private institutions.

Regarding the information systems side of management, there are a number of areas which will require special attention in Afghanistan to overcome the present technical focus on IT management:

- overall management of the information systems function in an organization (e.g. structuring of the IT/IS organization, management of the IT/IS workforce, relationship to the user community and to the organization's top management)
- development, implementation, and maintenance of computer-supported operational work processes and information systems
- development, implementation, and maintenance of computer-supported collaborative work processes and information systems (frequently summarized using the term “office automation”)
- development, implementation, and maintenance of basic IT/IS technical infrastructures (e.g. computers, data networks, basic network services, development environments, application systems).

Some of the functions above (e.g. overall information systems management) are genuine organizational functions and have to be fully integrated into the management structure of an organization. Other functions (e.g. application development and maintenance) will have to be outsourced in most of the cases because public organizations neither have (or should have) the staff resources and the required IT/IS expertise nor should they focus on topics which are foreign to their primary organizational functions. Furthermore, because of economies of scale factors and the shortage of qualified IT/IS staff in the Afghan public sector, outsourcing of IT/IS functions will be mandatory for most public organizations.

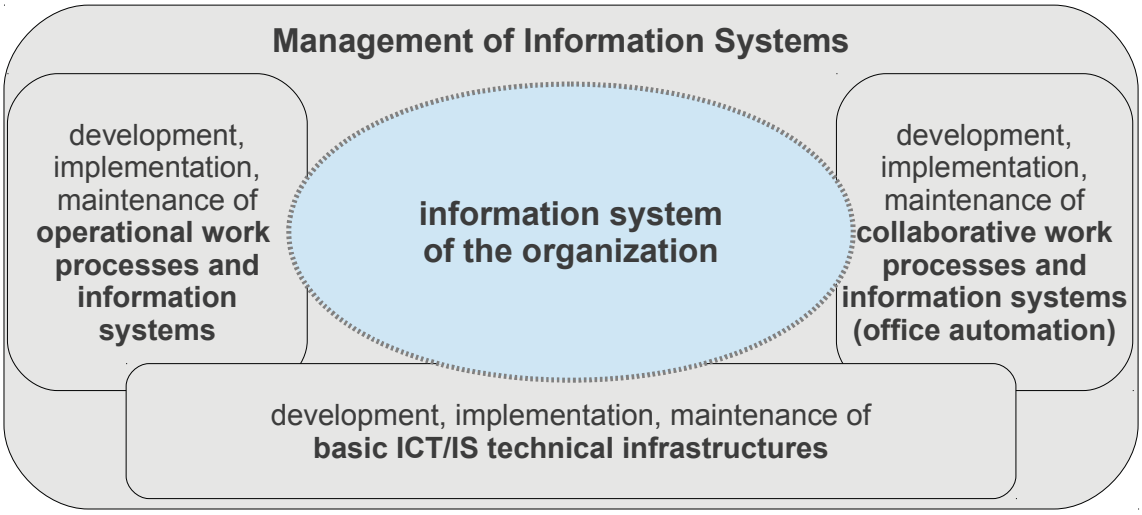


Fig. 1: Management of information systems – basic components

When we start to discuss the Open Source Movement and the term Open Source software (OSS) below we should keep in mind that Open Source systems can be used to build up or complement the system and application software layers of computer-supported information systems. But much more is required to put together viable organizational information systems.

The utilization of OSS in Afghan public institutions will be beneficial but it will not save the top management of these institutions from taking care of a broader set of information management issues as well – notably setting up suitable organizational structures for the information systems function of the organization and systematically developing computer-supported work processes for the most important services the organization is expected to deliver.

The present focus in many organizations on the management of technical IT issues might signal a general and severe competency problem as well: At Afghan universities there is a singular focus on introducing Computer Science faculties and the respective programs of study and there is not even one university program for educating Business Information Systems specialists at public universities.

1.2 Open Source software and the Open Source Movement

1.2.1 Open Source Software

The sharing of technical information is nothing new and the distribution and sharing of source code dates back to the early days of the Internet and the use of Bulletin Board Systems. [WikiP-02] It is reported that IBM made source code available for its operating systems and other programs as early as in 1955. [WikiP-02] Richard Stallman founded the Free Software Foundation in the middle of the 1980s and pushed the idea of “software freedom for all”. [WikiP-03] The term Open Source Software was coined in the late 1990s and the Netscape Navigator and the Linux operating system are prime examples of early Open Source systems. [WikiP-02, WikiP-03] In practical terms, there are little differences between “free software” and “open source” software: “... nearly all free software is open source, and nearly all open source software is free”. [GNU-01, GNU-02]

In 1998 “The Open Source Definition” was published by the Open Source Initiative and postulates the following: [WikiP-04, OSI-2012]

> Free redistribution, free access to source code, permission to produce derived works, integrity of the author's source code, no discrimination against persons or groups or fields of endeavor, distribution of product-independent license, license must be technology-neutral and must not restrict other software.

While there are different approaches of licensing OSS, the GNU General Public License is the prevalent scheme to license Open Source software. [WikiP-05, GNU-02, OSI-2012c]

1.2.2 Open Source Movement

Frequently, for Afghan people not familiar with the Open Source movement it is quite surprising that there is no business which sells Open Source software and frequently there is the question why it is not possible to buy the Linux operating system or other Open Source software in the bazaar.

While it might be hard to imagine - in a world which is focused on copyright wars (e.g. at present between Apple Inc. and Samsung on the mobile phone and tablet market) or extracting the maximum amount of money from hardware and software products – there are thousands of highly skilled altruistic programmers who volunteer in their free time to collaborate and produce high quality software and make this software available for free on the Internet. [WikiP-06]

The roots of the Open Source movement go back to the late 1970s and early 1980s when Richard Stallman created the GNU project and the Free Software Foundation on the US east coast and the Computer Science Research Group of the University of California at Berkeley made the BSD Unix code for i386 processors available free of copyrighted code. [WikiP-06]

Wyllys defines the Open Source movement as: “a world-wide movement composed … of many people who feel that the best way to produce software ... is to enlist the cooperation of interested, skilled, altruistic programmers who are willing to work for free, inspired by the twin goals of producing high-quality programs and of working cooperatively with other similarly minded people.” [Wyllys-2000, WikiP-01, WikiP-07]

Providers of traditional proprietary software systems did not necessarily welcome the Open Source movement and in 2001 Microsoft's Steve Ballmer was quoted to have compared Linux with cancer: “Linux is a cancer that attaches itself in an intellectual property sense to everything it touches ...”. [Greene-2001] In the case of Microsoft and from a marketing perspective this attitude towards Open Source software was not sustainable and the software vendor – while acting differently - now pretends: “We love open source.” [Brodkin-2010, Noyes-2012, Phipps-2012]

Today the Open Source movement can look back on a track record of highly successful projects like the Linux operating system, the Apache web server, the MySQL relational database manager, the LibreOffice office suite, the PHP scripting language, the Mozilla browser, and many others. [WikiP-06] While the large-scale penetration of the market for end-user PCs has not yet been accomplished, Linux and other Open Source software are well established in major IT/IS segments:

- Linux rules the supercomputer market segment: 91% Linux systems and 1.2 % MS Windows systems [Vaugh-2011a, Vaugh-2012]
- Linux is the prime choice of ISPs and the market share of Microsoft IIS dropped to under 12% [Pogs-2012, TechR-2012a, TechR-2012b, WikiP-08]
- in 2008 and after a disastrous crash of its MS Windows systems, the London Stock Exchange moved to Linux; the Qatar Exchange moved to Linux in 2010; the New York Stock Exchange made the move as did XETRA Stock Exchange [Vaugh-2011b]
- President Putin gave order to put the Russian government on Linux by 2015 [Gross-2010]
- the British government in 2012 passed regulations which will strongly encourage central government agencies to acquire Open Source systems [HMG-2012, FSFE-2012-b , Offerman-2012]
- … but it is reported that Linux on desktop systems only has a market share between 1% and 1.5% while Microsoft's market share is above 86% - which is put into question by other sources. [TechR-2012a, TechR-2012b] The strong growth of the Linux-bases Android operating system on mobile phones and tablet computers is threatening Microsoft's dominance in end-user computing at present: “Android had a 33 percent vendor share last year of the 'consumer computer' market to Microsoft's 25 percent.” [Poeter-2012]

Looking at these numbers, we have to take into consideration that Open Source software is available for Microsoft Windows operating systems, too. The global market share for Mozilla Firefox was reported at the end of 2012 to be at about 21.9% (Google Chrome about 36.4%, MS Internet Explorer about 30.8%). [StaCount-2012] According to StatCounter Global Stats, Microsoft with its Internet Explorer has lost the top position in the browser market since May 2012 to Google Chrome and the Open Source Mozilla Firefox is the long-term number three. [StaCount-2012]

In 2010 the Gartner Group estimated that more than 50% of IT organizations were using Open Source software. [Wilcox-2010]

1.2.3 What are the potential benefits?

Frequently, the major benefit which is attributed to Open Source software is the absence of licensing fees. But there is much more to mention [Wilson-2012, Montpel-2012, Noyes-2010]:

- the source code is always available and can be re-used or application systems can be customized or enhanced by a user organization (or their service providers)
- bug-fixing can be done easily and fast by anyone who has the appropriate skills or by the global Open Source community
- greater uptime, less downtime of Open Source systems
- software is based on standards and therefore interoperability is better
- security flaws (or the much rumored back-doors in proprietary systems) can be detected easily or – in the case of back-doors – will not be implemented in Open Source software at all
- the quality of the software is superior
- end-user interfaces can be translated as required
- vendor lock-in strategies which can lead to high switching cost can be avoided
- commercial software vendors can discontinue products or go out of business – because the code of an Open Source software system is not owned by anyone, the risks are lower
- users and developers can join Open Source communities and benefit from their activities

But … it has to be taken into consideration that the total cost of ownership of software systems consists of more than just licensing cost. Additional cost factors are: training, programming/customization, maintenance, or consulting and these cost factors cannot be eliminated by utilizing OSS. [GBDIR-2012]

1.3 Utilizing Open Source systems in Afghanistan

Utilizing Open Source systems will not solve every information system problem in Afghan public organizations. As we already have pointed out above, there is a

plethora of problems in the areas of professional IT/IS management, systematic development/implementation of work processes, education of end-users and IT/IS professionals, building up viable network infrastructures with sufficient bandwidth to connect geographically dispersed organizations and to communicate with the outside world, and to create/maintain a viable database and computing infrastructure. These problems will have to be solved alongside the introduction of OSS to make IT/IS work well in Afghan organizations.

1.3.1 What is the status of IT/IS in public organizations in Afghanistan?

At present the (rudimentary) use of Open Source IT/IS systems and infrastructures in Afghanistan's public institutions is mostly confined to a small number of leading public universities. The general knowledge how to utilize Open Source application systems and IT infrastructures seems to be minimal and up to now a competence center for Open Source systems and infrastructures which could provide professional training, consulting, and technical support to the public sector is missing.

While in many parts of the world governments, universities, city administrations, and large private organizations have already converted their applications and IT infrastructures to Open Source systems – notably Prime Minister Putin of Russia has ordered the complete Russian federal sector to convert its systems to Linux/Open Source software by 2015 – most public sector managers and IT directors in Afghanistan seem not yet to be aware of the opportunities an Open Source strategy holds for their organizations.

At the same time many public organizations in Afghanistan are in need to develop and deploy new information system infrastructures from scratch and by default (or lack of knowledge) will turn to proprietary operating systems and application software. Proper license agreements for the software in use (or to be used) might not be in place, software maintenance might not be possible, and especially end-user systems are prone to viruses or running boot-legged software which is not fully functional. In worst-case scenarios (which exist in Afghan organizations today) the hard-disks of PCs have to be re-formatted weekly to get rid of at least some of the viruses.

Based on these assumptions and observations, the professional implementation and use of computer-supported information systems seems to be hardly possible with the present approach, while at the same time there is an urgent need to build up information systems in many of the larger public organizations in Afghanistan.

In addition, a strategy to use (costly) proprietary software systems to develop information system solutions will not be sustainable in Afghanistan and information system solutions built on proprietary software might have to be scrapped or redesigned using Open Source software in the future – loosing all or most of the investments made during the present phase of IT/IS infrastructure build-up.

1.3.2 What could be done to enable a full utilization of OSS?

Due to the situation outlined above, it is urgently required to initiate a discussion with potential stakeholders in general (fig. 2) and with top managers of Afghan public institutions – and not only with IT managers - about utilizing Open Source systems, enhancing the competencies/capacities of IT units, and jointly develop strategies to introduce Open Source systems in Afghan public organizations.

An Open Source competence center in Kabul needs to be established urgently. This center – the proposed ACOSS-PO - Afghan Center for Open Source Systems in Public Organizations – would be able to transfer the required Open Source knowledge and competencies to Afghanistan and could provide professional education/certification, consulting, and IT/IS services to public organizations which want to set-up Linux- and Open Source-based information systems. [Finke-2012]

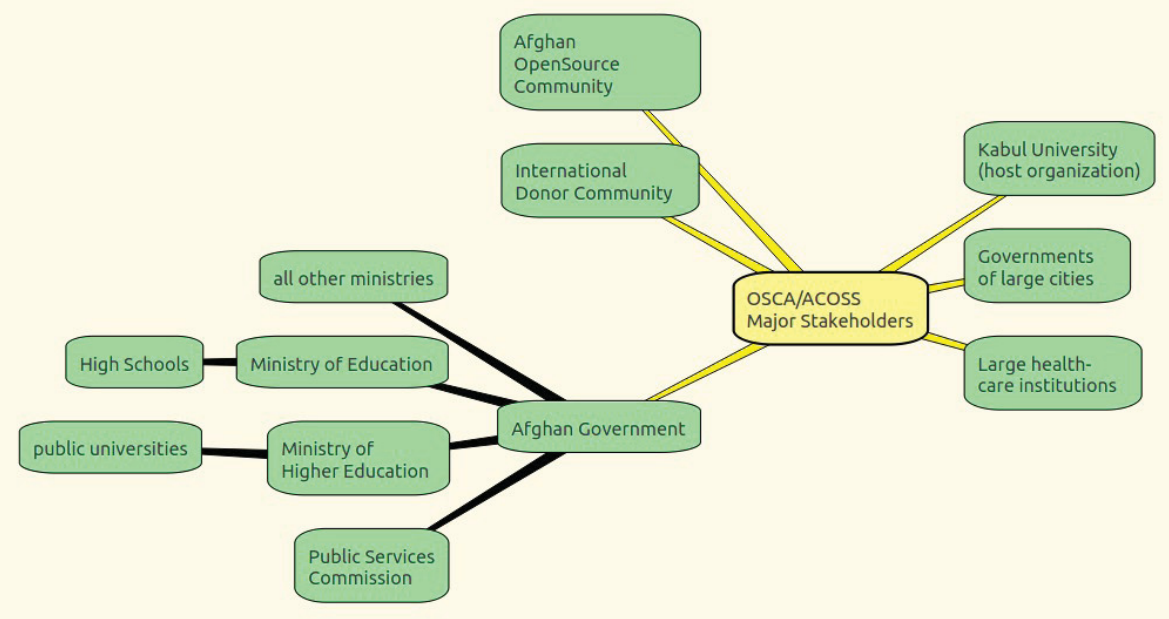


Fig. 2: Potential stakeholders of ACOSS-PO

In discussions, Afghan partners, IT/IS professionals, or members of the fledgling Afghan Open Source community (e.g. NICTAA National ITC Alliance of Afghanistan, TechWomen Afghanistan) frequently point out that the lack of professional knowledge, competencies, and services in Afghanistan are the prime inhibitors for the utilization of OSS. [TechW-2012, NICTAA-2012] At present, the availability of certified professionals in Kabul is a crucial factor when decisions about the utilization of OSS in government organizations are made. There are already certified IT/IS professionals available to support proprietary software products but there is hardly anyone (or none) available to provide professional and reliable long-term services for OSS. So, the results of the decision processes are predictable: Afghan public organizations are forced to adopt IT/IS infrastructures now which they will hardly be able to afford or maintain in the long range.

To break the cycle of available OSS knowledge/competencies, decisions which are leading to non-sustainable IT/IS systems and the resulting lack of business opportunities for OSS services providers, an Afghan Center for Open Source Systems in Public Organizations is mandatory: It is proposed to set-up a non-profit organization with the help of international donor agencies and in close co-operation with Afghan partner institutions (e.g. Ministry of Communication and

Information Technology, University of Kabul, National ITC Alliance of Afghanistan). The proposed OSS competence center should receive support for about three to five years and be able to earn the budgets for its operation by providing for-fee consulting, educational and IT/IS services to public organizations afterwards.

The quality of professional staff members and the services offered by ACOSS should be on par with international services providers but the organization should not be allowed to compete with private Afghan IT/IS services providers on price. Instead, it should actively support the development of private Afghan services and consulting businesses.

In general the organization should focus on the following goals:

- internal capacity building (educating, maintaining and developing a group of IT specialists, trainers, and consultants - including the capacity to professionally manage IT/IS projects)
- external capacity building (training and certification programs for the Afghan government and public institutions)
- providing professional IT, IS, and consulting services for the Afghan government and public institutions in Afghanistan
- providing general support (e.g. training, certification, business development support) for an emerging sector of private Afghan OSS service providers
- supporting/coordinating a lively Afghan Open Source community and systematically connecting it to international Open Source activists and groups.

In addition, there seems to be a chance to assign product specific competence center functions to ACOSS (e.g. second level support for OSS health management information systems proposed by the University of Oslo for larger Afghan hospitals).

As a matter of course, ACOSS should be modeled as an Afghan organization right from the start – in particular one that commands capacities on par with international IT/IS and consulting organizations and excels in applying its skills in an Afghan cultural/organizational environment. The following steps are pro-

posed to set-up the ACOSS organization:

1. initialization phase
   - establishing the legal framework and cooperation agreements
   - establishing an office/lab environment
   - acquisition of professional staff resources (3 IT/IS trainers, 5 IT infrastructure specialists, 5 IS specialists, 5 IS management and workflow consultants)
   - acquisition of IT/IS resources
2. preparing for doing business
   - education of professional staff resources up to international certifications
   - project management/administration education for managers (international certificates) and other staff members
   - setting-up lab facilities for training courses
   - development of IT/IS training courses, course materials, Afghan certification concepts
3. starting basic business operations
   - IT infrastructure consulting projects (limited complexity)
   - IS consulting/implementation projects (limited complexity)
   - IS management and workflow projects (limited complexity)
   - training courses (selected topics)
4. review/evaluation of phase of basic business operation
5. assuming full business operations

As stated above, ACOSS should have a strong focus on educating Afghan IT/IS specialists up to international IT/IS consulting and service providers' standards (phase 2 – preparing for business). It is intended to accomplish this mainly by providing training courses (conducted by international specialists) in Afghanistan (four to six weeks per training unit).

After a phase of about 12 to 18 months of intensive professional training of professional staff members ACOSS should enter a phase of “basic business operations” for another 12 to 18 months. During this phase ACOSS professionals

should conduct projects of limited complexity at clients' sites. These projects should be closely guided and supervised by experienced international project managers.

A phase of review and evaluation will follow (phase 4 – review/evaluation of phase of basic business operation). During this phase (about two months) projects will be evaluated in detail, “lessons learned” will be discussed, and required changes will be implemented. Afterwards ACOSS will assume full business operations and a review and evaluation cycle of twelve months will follow for three years.

The support of the Afghan Open Source community should be an ongoing process which could be initiated in phase two.

While the implementation of the ACOSS project will require considerable (but decreasing) up-front investments over a period of three to five years the organizational, technical and financial benefits of the project for Afghan public organizations would be huge and many times compensate for the initial investments and efforts.

References

[Brodkin-2010] Brodkin, Jon (2010): Microsoft: “We love open source”, in: Network World, Aug. 23, 2010, http://www.networkworld.com/news/2010/082310-microsoft-open-source.html, retrieved on Sept. 3, 2012.

[Finke-2012] Finke, Wolfgang F. (2012): OSCA Initiative - Open Source Computing in Afghanistan, Afghan Center for Open Source Systems in Public Organizations, http://www.winf.bw.fh-jena.de/www/winf-web.nsf/id/Forschung_DE, retrieved on Dec. 31, 2012.

[FSFE-2012-a] Free Software Foundation Europe (ed. 2012): UK takes major step towards competition, innovation in software market, http://fsfe.org/news/2012/news-20121101-02.sl.html, retrieved on Dec. 31, 2012.

[FSFE-2012-b] Gerloff, Karsten (2012): UK takes huge step forward on Open Standards, http://blogs.fsfe.org/gerloff/2012/11/01/the-uks-new-open-standards-policy/, blog retrieved on Dec. 31, 2012.

[GBDIR-2012] GBDIRECT (ed. 2012): Benefits of Using Open Source Software, http://open-source.gbdirect.co.uk/migration/benefit.html,

retrieved on Dec. 31, 2012.
[GNU-01] GNU/FSF (ed. 2012): GNU Operating System – Categories of free and nonfree software, http://www.gnu.org/philosophy/categories.html, retrieved on Aug. 31, 2012.
[GNU-02] GNU/FSF (ed. 2012): GNU Operating System – Why “Free Software” is better than “Open Source”, http://www.gnu.org/philosophy/free-software-for-freedom.html, retrieved on Aug. 31, 2012.
[Greene-2001] Greene, Thomas C. (2001): Ballmer: “Linux is a cancer” Contaminates all other software with Hippie GPL rubbish, http://www.theregister.co.uk/2001/06/02/ballmer_linux_is_a_cancer/
[Gross-2010] Gross, Grant (2010): Putin to put Russian government on Linux by 2015, on: Computerworld, http://www.computerworld.com/s/article/9202638/Putin_to_put_Russian_government_on_Linux_by_2015, retrieved on Sept. 09, 2012.
[Heise-2012] H-Open (ed. 2012): Firefox number two in global desktop browser share, http://www.h-online.com/open/news/item/Firefox-number-two-in-global-desktop-browser-share-1631559.html, retrieved on Sept. 9. 2012.
[HMG-2012] HMGovernment (ed. 2012): Open Standards Principles - For software interoperability, data and document formats in government IT specifications, http://www.cabinetoffice.gov.uk/sites/default/files/resources/Open-Standards-Principles-FINAL.pdf, retrieved on Dec. 31, 2012.
[Montpel-2012] Montpelier Open Source (ed. 2012):The Benefits of Open Source, http://www.montpelieropensource.com/benefits.htm, retrieved on Dec. 31, 2012.
[NICTAA-2012] National ITC Alliance of Afghanistan – http://www.nictaa.af.
[Noyes-2010] Noyes, Katherin (2010): 10 Reasons Open Source Is Good for Business, http://www.pcworld.com/article/209891/10_reasons_open_source_is_good_for_business.html, retrieved on Dec. 31, 2012.
[Noyes-2012] Noyes, Katherin (2012): Microsoft Delivers a Blow to Open Source with Visual Studio 11, in: PC World, May 27, 2012, http://www.pcworld.com/businesscenter/article/256339/microsoft_delivers_a_blow_to_open_source_with_visual_studio_11.html , retrieved on Sept. 3, 2012.
[Offerman-2012] Offerman, Adrian (2012): Issues in open source procurement in the European public sector I, published via European Commission Joinup, https://joinup.ec.europa.eu/software/studies/issues-open-source-procurement-european-public-sector-i, retrieved on Dec. 31, 2012.
[OSI-2012a] Open Source Initiative (ed. 2012): The Open Source Definition, http://opensource.org/docs/osd, retrieved on Aug. 28, 2012.
[OSI-2012b] Open Source Initiative (ed. 2012): The Open Source Definition,

http://opensource.org/docs/osd, retrieved on Aug. 28, 2012.
[OSI-2012c] Open Source Initiative (ed. 2012): Open Source Licenses, http://opensource.org/licenses/index.html, retrieved on Dec. 31, 2012.
[Phipps-2012] Phipps, Simon (2012): The real reason behind Microsoft's leap into open source, in: Infoworld, April 20, 2012, http://www.infoworld.com/d/open-source-software/the-real-reason-behind-microsofts-leap-open-source-191251, retrieved on Sept. 3, 2012.
[Poeter-2012] Poeter, Damon (2012): Goldman Highlights Microsoft's Shrinking Market Share, http://www.pcmag.com/article2/0,2817,2412997,00.asp, retrieved on Dec. 31, 2012.
[Pogs-2012] Pogson, Robert (2012): Netcraft: M$ Drops to Number Three on Active Websites, http://mrpogson.com/2012/08/03/netcraft-m-drops-to-number-three-on-active-websites/, retrieved on Sept. 9, 2012.
[TechR-2012a] Techrights (ed. 2012): GNU/Linux Market Share, http://techrights.org/wiki/index.php/GNU/Linux_Market_Share, retrieved on Sept. 9, 2012.
[TechR-2012b] Techrights (ed. 2012): True Desktop Market Share for GNU/Linux, http://techrights.org/2012/03/05/tux-share/, retrieved on Sept. 9, 2012.
[TechW-2012] TechWomen Afghanistan – http://www.techwomen.af.
[StaCount-2012]. .StatCounter Global Stats – Top 5 Browsers from Dec. 2011 to Dec. 2012, http://gs.statcounter.com/, retrieved on Dec. 31, 2012.
[Vaugh-2011a] Vaughan-Nichols, Steven J. (2011): The Best & Fastest Computers are Linux Computers, on: ZDNET, retrieved on Sept. 9, 2012.
[Vaugh-2011b] Vaughan-Nichols, Steven J. (2011): The London Stock Exchange moves to Novell Linux, on: ZDNET, http://www.zdnet.com/blog/open-source/the-london-stock-exchange-moves-to-novell-linux/8285, retrieved on Sept. 9, 2012.
[Vaugh-2012] Vaughan-Nichols, Steven J. (2012): Fast, Faster, Fastest: Linux rules supercomputing, on: ZDNET, http://www.zdnet.com/blog/open-source/fast-faster-fastest-linux-rules-supercomputing/11263, retrieved on Sept. 9, 2012.
[Wilcox-2010] Wilcox, Joe (2012): Gartner: More than half of IT organizations deploy open-source software, http://betanews.com/2011/02/08/gartner-more-than-half-of-it-organizations-deploy-open-source-software/, retrieved on Dec. 31, 2012.
[WikiP-01] Wikipedia (ed. 2012): Open Source, http://en.wikipedia.org/wiki/Open_source, retrieved on Aug. 28, 2012.
[WikiP-02] Wikipedia (ed. 2012): Open source software development, http://en.wikipedia.org/wiki/Open-source_software_movement, retrieved on Aug. 28, 2012.
[WikiP-03] Wikipedia (ed. 2012): Free Software,

http://en.wikipedia.org/wiki/Free_software, retrieved on Aug. 28, 2012.

[WikiP-04] Wikipedia (ed. 2012): The Open Source Definition, http://en.wikipedia.org/wiki/Open_Source_Definition, retrieved on Aug. 28, 2012.

[WikiP-05] Wikipedia (ed. 2012): GNU – General Public License, http://en.wikipedia.org/wiki/GNU_General_Public_License, retrieved on Aug. 31, 2012.

[WikiP-06] Wikipedia (ed. 2012): Open source movement, http://en.wikipedia.org/wiki/Open_source_movement, retrieved on Sept. 3, 2012.

[WikiP-07] Wikipedia (ed. 2012): Open source, http://de.wikipedia.org/wiki/Open_Source, retrieved on Sept. 3, 2012.

[WikiP-08] Wikipedia (ed. 2012): Web server, http://en.wikipedia.org/wiki/Web_server#Market_share, retrieved on Sept. 9, 2012.

[Wilson-2012] Wilson, James A. J.(2012): Benefits of open source code, http://www.oss-watch.ac.uk/resources/whoneedssource, retrieved on Dec. 31, 2012.

[Wyllys-2000] Wyllys, R. E. (2000): Overview of the Open-Source Movement, http://www.ischool.utexas.edu/~l38613dw/readings/Open SourceOverview.html, retrieved on Aug. 28, 2012.

2. Open Source Systems in Health Care Management

Michael Weinhara

EPOS Health Management GmbH, Bad Homburg/Germany

The health care system in Afghanistan suffers in the area of Health Information from a patchwork of applications, with little central coordination. The core area of the national Health Information System (HIS) focuses on data for internal facility administration, central allocation and management of resources. Open source systems play no significant role at health provider level.

2.1 Routine Data

For the collection of required public health statistical data, similar to that published by WHO or World Bank, a routine data stream from local health care providers has been established over the years. The data is still not as reliable in quality or completeness.

Performance data management for quality of care and cost control is still at an early or experimental stage and not yet acceptable as serious instrument for managerial decisions or performance based resource allocation. Important statistical data are therefore still often the result of specific surveys rather than the result of systematic routine data collection and analysis.

2.2 Stakeholders

Officers in charge for health information are often young and committed doctors, but unfortunately there is frequent change in those positions. The high frequency of change, also at higher positions – up to the deputy minister level, hampers progress in development. Briefings to new stakeholders are often required as internal hand-overs at all levels of the health system, fall short in structure and completeness in terms of IT. New heads of directorates, provincial health departments or hospitals also often wish to re-negotiate previous agreements.

2.3 Corruption

Corruption at all levels of health care provision is a significant problem and involves individuals throughout the hierarchy of health services. With a rank of 176 from 178 listed countries, Afghanistan is rated as the third most corrupt country globally, by Transparency International in 2010. A functional HIS, which provides reliable data, is not appreciated by these corrupt structures. The resource allocation in the health sector is generally insufficient. The sector also lacks of adequate instruments for monitoring.

2.4 World Bank Support

The World Bank financed Hospital Reform Project has improved the situation in selected provincial hospitals over the last years and started monitoring exercises, where quality of provided data is controlled by auditors during on-site visits. Even if reliable performance data are available, typical managerial decisions based on such information are often influenced by informal structures, unpredictable budget cuts or simply not done. This is because stakeholders have no capacity to interpret data for their decision making process. The mentioned World Bank projects finalize in 2013 although they would require more time to stabilize achievements

2.5 Quality

Even simple and basic paper-based documentation in health records and admission discharge lists are unreliable, as staff are not sufficiently trained in use. Most of the leading doctors of public hospitals run private hospitals in parallel. While public hospitals are officially not allowed to charge fees, they are chronically under-financed. As consequence, public hospitals are often used as recruitment centers for patients to private hospitals, rather than to provide acceptable medical services. Since many groups in public sector health facilities benefit from this desolate situation, change cannot be expected soon. Neither will change be easy.

2.6 Health Indicators

Afghanistan's health data is dramatic in terms of critical indicators. With a ratio of 0.1 hospital beds per 1000 citizens population (World Bank, 2011), and an infant mortality in the first year of life, of 103 deaths per 1000 live births (WHO, 2011) and an average life expectancy of 48 years, the Afghan health indicators range worldwide among the worst of all countries.

2.7 Coding Systems and Software

Medical classification and coding systems such as ICD-10 or ICPC-2 are not introduced in public facilities. Together with the known infrastructure and security problems, the knowledge and motivation of health care staff to provide accurate data is accordingly low. Applications in use are built generally on MS Access and Excel, frequently using illegal copies. As Iranian versions of Microsoft products with cracked product keys are available on all bazaars over the country for about 2 USD, there is no incentive for using other products. Among users, there is little problem awareness on illegal software usage. Capable Afghan IT experts, with in profound knowledge on databases, are difficult to find and the majority are only familiar with, and focused on Microsoft products.

2.8 HIS at Central Level

The central administration at Ministry of Public Health (MoPH) runs a department for information technology and receives support from the USAID funded Medical Science for Health (MSH). This organization has revised, over the last years, the Health Management Information System procedures manual, forms, and training-of-trainer guidelines (USAID, 2011) at central level. The software used for collection of pre-aggregated data is mainly build on MS software, i.e. Access. Among databases for collection of pre-aggregated data, a slim Electronic Medical Record solution is part of the results and made available at the MoPH. The applications offer hospitals with available resources, the option to make a substantial step towards IT based solutions in health care, but all require an MS environment. The main problems with the illegally copied software are the frequent loss of system stability, leading to loss of data; the absence of updated virus protection; inappropriate use of IT workplaces and theft of hardware. This all adds to the problems in HIS.

2.9 Open Source Solutions

The range of available and internationally well tested open source applications, like DHIS-2 and open EHR could cover most of the needs of the afghan health sector and increase system stability. To introduce OSS for an improvement of HIS would require that the applications are introduced to routine level usage in several pilot sites with authorization of the MoPH. This would allow stakeholders to observe how it operates in real working conditions. As the country suffers from a low level of education in general and even the educational elite at MoPH level have little or no experience in fully understanding how real health systems work it is clear that onsite demonstration is a good way forward.

It would be too much to expect that solutions presented in concepts, are understood by the client, unless a pilot HIS can be observed as part of the daily working routine.

If open source solutions in the health sector should be promoted, it would require starting with reference facilities where real implemented solutions allow practice. This approach would help the MoPH to develop its national concept

based on real facts. Any such approach would require considerable resources and time. Both is regularly underestimated by the big international donor agencies.

References

http://afghanistan.usaid.gov/en/USAID/Activity
http://data.worldbank.org/indicator
http://dhis2.org/
http://www.openehr.org
http://www.transparency.org/cpi2010/results
http://www.who.int/healthmetrics
http://apps.who.int/ghodata/?vid=3000&theme=country

3. Potential for Afghan participation in a global FOSS community of the health sector: National, regional and global opportunities

Sundeep Sahay and Johan Sæbø

University of Oslo, Oslo/Norway

For the conference “Creating Awareness for the Use of OpenSource Systems in the Public Sector in Afghanistan”, Kabul 2012, the authors were invited to give two presentations; One on the open source community called Health Information Systems Programme (HISP), and the other on HISP’s experience in creating regional networks and the potential for this in Central Asia with a particular focus on Afghanistan. Following these presentations, we engaged in discussions with representatives from the Ministry of Health of Afghanistan, who are considering porting their current health management information system (HMIS) from an MS Access platform, to open source. This paper follows up on these discussions, outlining the topics of our presentations as well as the initial work and future plans for an open source HMIS for Afghanistan.

3.1 HISP: A global community around open source health information systems

HISP is a network of health information researchers and practitioners spanning more than 20 countries. HISP consists of formal entities, such as universities, NGOs, Ministries of Health, and more informal participants such as idealists and individuals. HISP started as a initiative coordinated by the University of Oslo in

the mid-nineties in South Africa after the break down of apartheid, and today has expanded to many more countries in Africa, Asia, and recently in Latin America. The map below gives an indication of which countries is HISP active with currently [see www.dhis2.org].

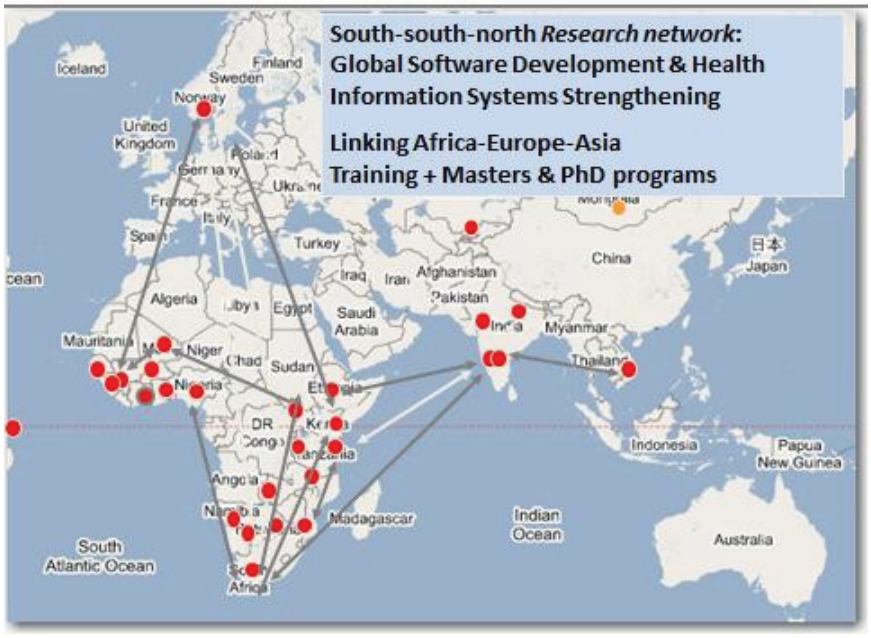


Fig.1: DHIS2 spread globally

The key "organizing" element in the HISP network has been the continuous and longitudinal development and application of the open source software platform called DHIS - District Health Information System. The DHIS is a software tool for collection, validation, analysis, and presentation of aggregate statistical data, tailored to supporting integrated health information management activities, with the potential of it being adapted to serve other areas. It is designed to serve as a district based country data warehouse to address both local and national needs. DHIS is a generic tool rather than a pre-configured database application, with an open meta-data model and a flexible user interface that allows the user to design the contents of a specific information system without the need for programming. The DHIS is developed to serve as a flexible tool that can be easily tailored to particular needs and requirements.

DHIS development has evolved over two versions. The first – DHIS v1- was developed since 1997 by HISP in South Africa on MS Access, a platform selected

because it was at that time a de facto standard in South Africa [1]. However, its distribution was free. The second – DHIS v2 – is a modular web-based software package built with free and open source Java frameworks, developed since 2004 and coordinated by the University of Oslo. The version 2 software has used as its key requirement specification the version 1, and to that has added now much functionality, driven by user requirements.

Over the years, HISP has evolved as a network, which we can loosely term as –Free and Open Source Network around Health Information Systems (FOSNHIS). This network represents a collective of entities linked together and working towards a common purpose – in our case strengthening public Health Information Systems (HIS). HIS represents an umbrella term including different information systems for the health sector such as Health Management Information Systems, patient based systems, HR systems, etc, In each case, the network deals with free and open source software where there are no licenses associated, and source code is freely available to make modifications for local use.

Globally, now there are many such FOSNHIS available. While ***HISP*** – Health Information Systems Programme around the DHIS is one such network, coordinated by University of Oslo, there are others such as:

- ***OpenMRS*** – around patient record systems – OpenMRS – coordinated by Indiana University
- ***iHRIS*** – around Human Resources information systems – coordinated by Capacity Plus
- ***ELISA*** – around Laboratory Information Systems – coordinated by Washington University

Such networks are characterized by a common purpose, in our case the design, development, implementation and dissemination of HIS, with a particular focus on developing countries. An important logic by which these networks operate is of learning in collectives, so that one can learn and share from each other, and avoid reinventing the wheel. It is now increasingly being acknowledged that networks are the solution to the challenges of scale and sustainability of HIS and are even being facilitated by international agencies such as WHO, UN and HMN. One network cannot try to work with all kinds of HIS, so each should try

to develop their own where they have expertise, while there should be attempts to develop processes of interoperability such that systems can talk to each other. The WHO is leading the initiative to develop standards for such interoperability, and many are working on the development of '*Integrated Health Information Architectures*' within the framework of a health systems approach.

The guiding logic in HISP is of 'networks of action', in which the coordinating entity promotes action to strengthen collaboration and sharing between network members around software (in case of HISP it is DHIS), resource material such as training and user manuals, and also implementation experiences and best practices [2]. The action comes in many forms including the generation and sharing of funding, circulation of people, ideas and experiences across the network. A lot of circulation in the network takes place electronically, such as the use of mailing lists and web-pages. HISP can be seen to consist broadly of three sets of action. The first relates to the DHIS and its design, development and sharing.

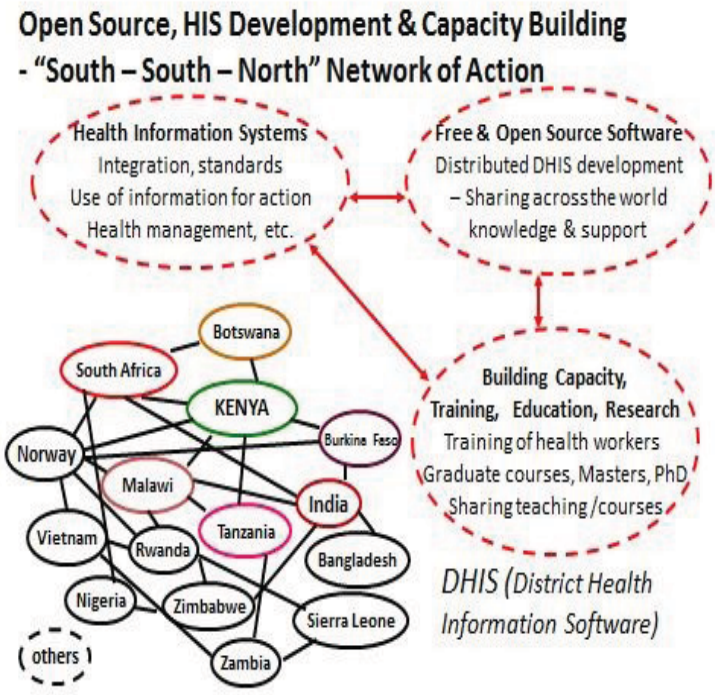


Fig. 2: The HISP network of action

The second consists of research and education focused on building capacity. This includes three levels of in-service training, masters education and doctoral research broadly coordinated by the Department of Informatics at Oslo in collaboration with various universities in the South. The third relates to supporting practical HIS implementation in countries including design of systems, their integration, support and various others. The HISP supported FOSNHIS can be depicted through figure 2.

The Afghanistan Ministry of Health can gain through joining a network such as HISP. They gain the experience around the design, development, and implementation of HIS which has been accumulated over the years, and to which they can also contribute. They can by engaging in the different activities and forums of the network, build specialized capacity and also gain latest developments in products like DHIS2, iHRIS and OpenMRS. They can draw upon the network resources to acquire and also provide services relating to technical support in setting up systems, customizing, maintaining, and upgrading them. The network provides a collective forum in which they can learn and grow with others in same situation, and also gain increased opportunities for attracting funding by aligning with global standards.

It is rather simple for someone to join the network, even just involving a simple email expressing interest to join. Since a key approach is on 'learning by doing', it helps to early on initiate processes to establish a pilot project, with a clear responsibility on who would coordinate the project which could be the Ministry, NGO or University. Joining electronic mailing lists – for developers, implementors, trainers etc – helps to quickly situate yourself within the sub-community of your interest. Within the framework of the project, you can engage in processes of customization of the application with others as a mechanism to learn. Once you can create a robust 'national HIS network', it will help to better link up and leverage more effectively the global FOSNHIS. For example, in HISP there are many different models in the development of the national network, including:

- NGOs – HISP India, HISP South Africa, HISP Vietnam, HISP West Africa, HISP East Africa, etc
- Universities – Dar-Es-Salam, Tanzania, Addis Ababa, Ethiopia etc
- Ministry of Health – Kenya, Malawi, Ghana Health Services

- Other networks – EHAS in Latin America, WAHO in West Africa – forming network of networks [3]

While FOSNHIS provides tremendous opportunities for engaging, learning and strengthening of HIS, there are multiple challenges too. A key one is to understand that free software is not free, and lots of effort and resources need to go into processes of customization, capacity building, and implementation efforts. In most developing countries, the public systems are not geared to managing procurement of open source software, so it is hard to find an institutional space to operate. A great deal of institutional capacity, including long term vision and guidance, is needed to help make the potential of open source realized in practice, especially within a public health system. Infrastructure and human resource issues required to deal with open source are non-trivial, and need long term planning. One of the strategies to deal with the challenges are 'regional networks' which we now discuss.

3.2 HISP Open Source community: hisp global, regional experiences (west Africa, Latin America)

In West Africa, HISP is today present in around 10 countries. In 2008, Sierra Leone was the first country outside India to implement DHIS2, as part of a project supported by Health Metrics Network [4]. As the project, and the first promising results, came to be known, more countries showed interest in also using DHIS2. The Gambia was the second country, and, like Sierra Leone, it had weak institutional capacity to efficiently manage a national data warehouse for health information. In the first years then, the global HISP community was constantly supporting the two implementations on the ground, a process that was costly and inefficient.

In 2010, interoperability between DHIS2 and the open source medical record system OpenMRS was established, and a first pilot was set up in Sierra Leone. Such interoperability was in high demand in the region, and the West African Health Organization (WAHO) organized for a demonstration at a regional conference to improve (health) human resource management. A third software, iHRIS, was being promoted by WAHO as an open source solution to HR man-

agement. This in turn led to plans to develop interoperability between all three, creating an open source suite of tools. Supported by WHO and HMN, this was officially launched at the end of 2010.

However, there were many challenges to implement this in the WAHO member states. On the technical side, there should be standards for both data definitions and data transmission. On the organizational side, few of the countries had the capacities to implement this suite of tools, let alone just one of the applications. The strategy to overcome both was to create a regional network, where WAHO would play a key coordinating role.

This regional collaboration can be outlined as three different streams:

First, country implementations of DHIS2, usually supported bi-laterally by the University of Oslo. Around 10 countries are now in the process of implementing DHIS2, and many face the same challenges. It is therefore a strategy to promote the emergence of local entrepreneurs to better meet the local demand. One company, HISP West Africa, has been established, and consultants attached to this company are increasingly able to offer support across the region.

Second, the leading role of WAHO at the regional level to coordinate and support the country implementations. For example, WAHO has developed a regional policy document for HIS, which all countries have approved. This document spells out the regional requirements relating to standards, but also helps the countries develop their own HIS policies.

Thirdly, the community building efforts through joint workshops. WAHO is holding annual HIS conferences on a more political level, while the DHIS2 Academy is about to be held for the third time. In this workshop, participants from all implementing countries gather for training, discussion, and learning from each other.

Another regional effort is just being established in 2012. Following interest from a handful of Latin American countries, a small group of institutions have done some preliminary work to translate DHIS2 to Spanish and set up pilots. This network, though only in its very beginnings, have also focused more on participation of universities in the implementing countries. This strategy is based on earlier experiences from other HISP countries, to build long-term capacity in the

country.

3.3 Promises of DHIS2 in Afghanistan

The current HMIS in Afghanistan is supported by a software using MS Access, which to our knowledge works well. However, due to the database limits of MS Access, only one year of data can be in the database. Archiving is done yearly, which complicates the analysis of historic data, assessments of trends, etc.

To counter this, the Ministry of Health has been looking at other databases that would be able to handle more data. We would like to point out a few other potential benefits of making a shift to open source software, in this case DHIS2.

- The benefits of a global network that is constantly adding functionalities and robustness of the software, and which is at the forefront of HIS development around the world
- Get on-line with a web-based system, reducing logistics related to updates and trouble-shooting and increasing access to data for all users

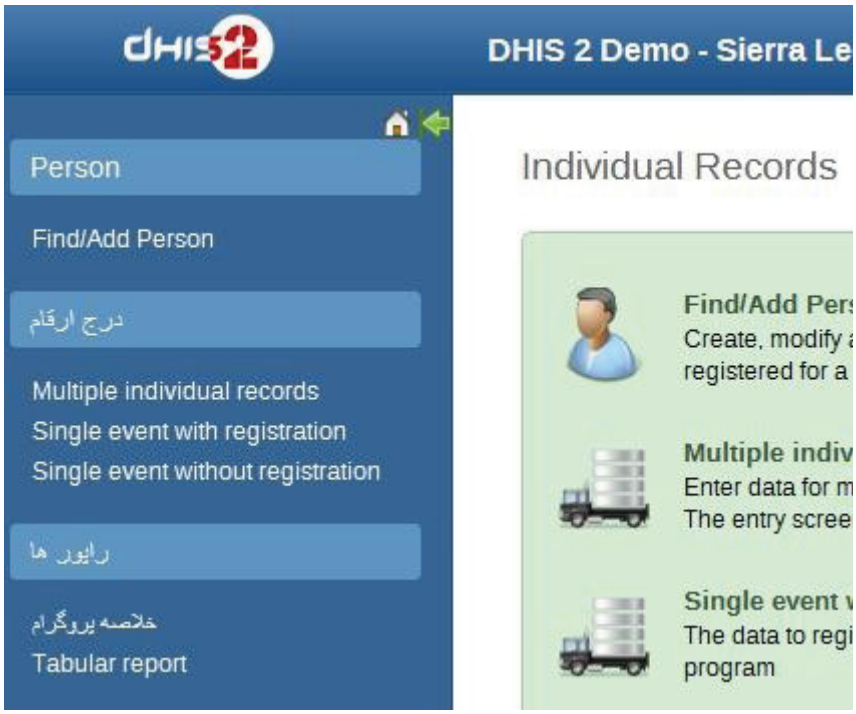


Fig. 3: DHIS2 screenshot

- Possibility to make modifications of the code
- Less vulnerable to virus
- Standardized training curricula and material, and yearly regional "DHIS2 Academy". For instance, Tajikistan is about to implement DHIS2 as a national data warehouse, after a pilot project [5]

Initial efforts has already been made to test the software in Dari and Pashtu. A developer in the global community has added the languages to the list of supported scripts, and a representative from Ministry of Health in Afghanistan has done some translation to the central translation server. The screen-shot above shows the first translated text in DHIS2. The software is already multilingual, so adding new languages is not a complicated procedure. However, as can be seen, there will be issues around aligning the text to the right of buttons and menus, even if the right to left script is already supported.

References

Braa, J. and C. Hedberg, The Struggle for District-Based Health Information Systems in South Africa. The Information Society, 2002. 18: p. 113-127.

Braa, J., E. Monteiro, and S. Sahay, Networks of Action: Sustainable Health Information Systems Across Developing Countries. MIS Quarterly, 2004. 28(3): p. 337-362.

Sæbø, J.I., et al. Networks of Networks - Collaborative Efforts to Develop Health Information Systems Across Developing Countries. in Ifip WG 9.4 Social Implications of Computers in Developing Countries. 2011. Kathmandu, Nepal.

Sæbø, J.I., et al., Comparing strategies to integrate health information systems following a data warehouse approach in four countries. Information Technology for Development, 2011. 17(1): p. 42-60.

Sahay, S., et al., Interplay of Institutional Logics and Implications for Deinstitutionalization: Case Study of HMIS Implementation in Tajikistan. Information Technologies and International Development, 2010. 6(3).

4. OpenSource Systems in Public Universities – Current state and some history

Frank B. Brokken

University of Groningen, Groningen/Netherlands

4.1 Introduction

Open source, when encountered first, seems like an amazing and maybe even impossible development. Open source software, software that comes not only with a ready-to-run binary program, but also with its defining program source code often raises serious questions and concerns.

- How is it possible that people and even organizations make their intellectual products available, allowing others (maybe even competitors) to use their products for free?
- How could open source products even start to compete with big software vendors when it comes to maintaining high quality standards?
- How could open source products even start to compete with big software vendors when it comes to offering support, including the proper handling of bug-reports?
- Why would anybody invest a lot of time and effort in developing software without even the slightest hope of receiving any financial compensation?

These and comparable questions are often raised when the topic of `open source' enters a discussion. And yet, almost 25 years after Richard Stallman initiated the `Free Software Foundation, we can conclude that Open Source soft-

ware, and its companion: open standards, has gained itself a strong position in the world of Information Technology. Much stronger, indeed, than most ordinary computer users are aware of.

What some may think of as a *fairy tale* isn't a fairy tale at all. It's a reality that came into existence through the combined efforts of thousands; people who thought that computer software should be a commodity, like the air we breathe.

In the early days of the open source movement, Eric S. Raymond, one of the main architects of the open source movement wrote down the movement's philosophy in his 1999 (2001) book *The Cathedral and the Bazaar.* When reading that book it becomes clear that open source software is a viable alternative to the ideas main software vendors have tried to instill in us, i.e., that software is expensive and proprietary, and that, because of that, we should turn to them for the software of our computers. Might it be that the ordinary computer user thinks this way because he/she learned that computers and their software are expensive. These days, however, those thoughts only seem to prevail at the *personal computer* level. When we look beyond that, a completely different picture emerges.

In this paper I'll give a short overview of the history of computers, thereby explaining why many ordinary computer users think computer software should be expensive. Next I'll turn to the origins of the Open Source movement, and to its current status. Finally you may ask yourself *why* you're using computers and whether you're required to use proprietary, closed source software at all. In many cases there *will* be a perfect alternative available.

4.2 Computers - Some history

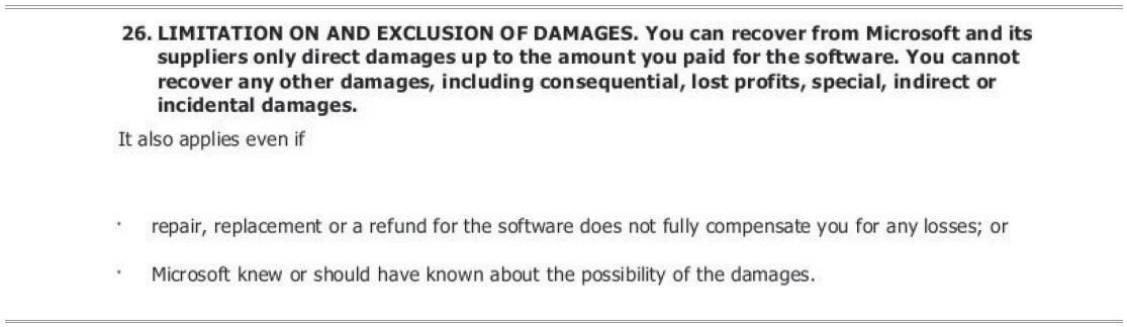
Between 1943 and 1945 computers started to emerge, mainly as the result of the efforts that went with World War II. These initial computers, like the *Electronic Numeric Integrator and Calculator (ENIAC)* were huge. They easily filled up a large living room, they were immensely heavy (e.g., they weighted over 30 tons), and were big power consumers (the ENIAC was rated at 174 kW). These computers were big, and with their many components (like over 18,000 vacuum tubes) also very expensive. Of course, you couldn't go to a software company

for their programs: developing the software took a lot of effort, the cost of which was calculated in man-years.

Several decades later we entered the era of the big mainframe computers. A well-known computer from the 1970s was the *Control Data Corporation (CDC) `Cyber'*, showing a price tag of some $20,000,000.-. Not something that was available at an ordinary educational or governmental institution. Its performance, those days, was impressive. Its clock ran at about 40 MHz, its working memory consisted of some 250,000 60-bit words. Of course, CDC closely guarded the secrets of its operating system, and when you bought the *Cyber* you usually also bought a maintenance contract, allowing *CDC*-technicians to perform software updates.

The lessons we all had learned from these big machines was that computers are expensive, and that they come with proprietary software, representing a non-significant fraction of the computer's price.

In 1981 *IBM,* up to then mainly active in the field of (really!) *business machines* and mainframe computers, decided to step into the *home computer* market, by then dominated by *Commodore, Atari,* and *Apple. IBM* introduced its *IBM-Personal Computer* on August 12, 1981. It had a 1MB physical address space and ran at a clock speed of 4.77 MHz. Unfortunately, IBM had no operating system for their latest product.



26. LIMITATION ON AND EXCLUSION OF DAMAGES. You can recover from Microsoft and its suppliers only direct damages up to the amount you paid for the software. You cannot recover any other damages, including consequential, lost profits, special, indirect or incidental damages.

It also applies even if

- repair, replacement or a refund for the software does not fully compensate you for any losses; or
- Microsoft knew or should have known about the possibility of the damages.

Fig. 1: Excerpt from the Microsoft end user-license

When they went shopping for an operating system, they met *Bill Gates* of the young Microsoft corporation, offering the IBM representatives not only a good

lunch, but also its *MS-DOS* operating system. In fact this was a bit premature, but Microsoft *bought* an operating system in 1981 from *Seattle Computer Products.* Lots of money again passed the counter, and eventually the user of the *IBM PC* and *MS-DOS* had to pay the bill (pun not intended).

At that time we had not only learned that we had to pay for computer software, but also that, as users of software, we only had one right: the right to pay the software vendor. Even today Microsoft's end-user agreements greatly limits its own responsibilities. Figure 1 shows an excerpt from Microsoft's actual end-user license.

Licenses like these quickly became the laughing stock of other manufacturers. In a reaction to Microsoft's license *General Motors* thought they would get away with selling cars that Sometimes just dies on the motorway for no reason. But for some reason the car owners simply accept this. They just restart and drive on....

4.3 The Beginning of Open Source

On October 4, 1985, *Richard Stallman* initiated the *Free Software Foundation (www.fsf.org).* As explained in their regulations, this doesn't necessarily refer to *price,* but to the *freedom to create,* distribute, and modify computer software.

Often this results in software which is free of charge. But also, and more importantly, creators of free and open source software typically have a personal interest in upholding its quality. In addition, since the source code *must* be available, thousands of technically trained people can look at the code and make concrete suggestions for improvements and bug-repairs. This situation is diametrically opposite from what we usually encounter with proprietary software, where bug reports usually seem to end up in the *big black hole*. This latter point is not amazing. Software vendors have no interest in improving the quality of their software, but primarily aim at profit maximization.

Open Source products are not without licenses. The *Free Software Foundation* promotes a very restrictive license: the *Gnu General Public License (GPL),* in fact implying that all products using this license themselves are subject to this GPL. I have no problems with this. In fact all my open source products (see,

e.g., *http://stealth.sourceforge.net*) are made available under the GPL. A more general approach is taken by the *Open Source Initiative (OSI, http://www.opensource.org).* The OSI offers a broad range of *open source licenses,* all of which require the software to be distributed together with their sources.

On October 1991 *Linus Thorvalds* released the first (beta) edition of his *Linux* operating system. Being a *Unix* variant it was a far cry from the *MS-DOS* system home computer users had grown accustomed to, but it was a relief to everybody else. On December 8 and 9, 1984 I organized, together with Karel Kubat and Piet Plomp, the *First Dutch International Symposium on Linux,* and Linus made some memorable statements during that symposium (like *We all know Linux is great, it does infinite loops in five seconds)*.

Since then, Linux has come a long way. It is not the only available free operating system, but it has pervaded many areas of computer use. It is available on more computer hardware platforms than any other operating system, almost all of the fastest supercomputers run Linux, it virtually owns the embedded system market (cell phones, tablet computers, network routers, the Android smart phone, etc, etc.)

After several decades of Open Source, we *could* have learned that

- Software code can be freely accessible;
- Software does not necessarily exist to maximize the vendor's profits;
- The standards of quality of open source products typically are very high
- Many systems in fact rely upon and use open source software
- Almost 80% of the Web-server market is occupied by open source products

Did the ordinary computer user become aware of all this? Let's briefly summarize the reasons why people are interested in open source products. Price, the availability of the source code, and the community review appear to be the main reasons (figure 2). However, in 2011 of 300 large organizations in the public and private sectors,

- 30% had decided to migrate mission-critical software to open source products;
- almost 70% had increased their investments in open source;
- 50% of the organizations were fully committed to the use of open source products;
- and almost 30% were about to take the step to using open source products.

In addition, many countries, as well as the United Nations had decided to either switch to, or strongly promote, the use of open source products and open standards (cf. document *100416_Open_Source_Policies.pdf,* published in 2010 by the *Center for Strategic and International Studies.)*

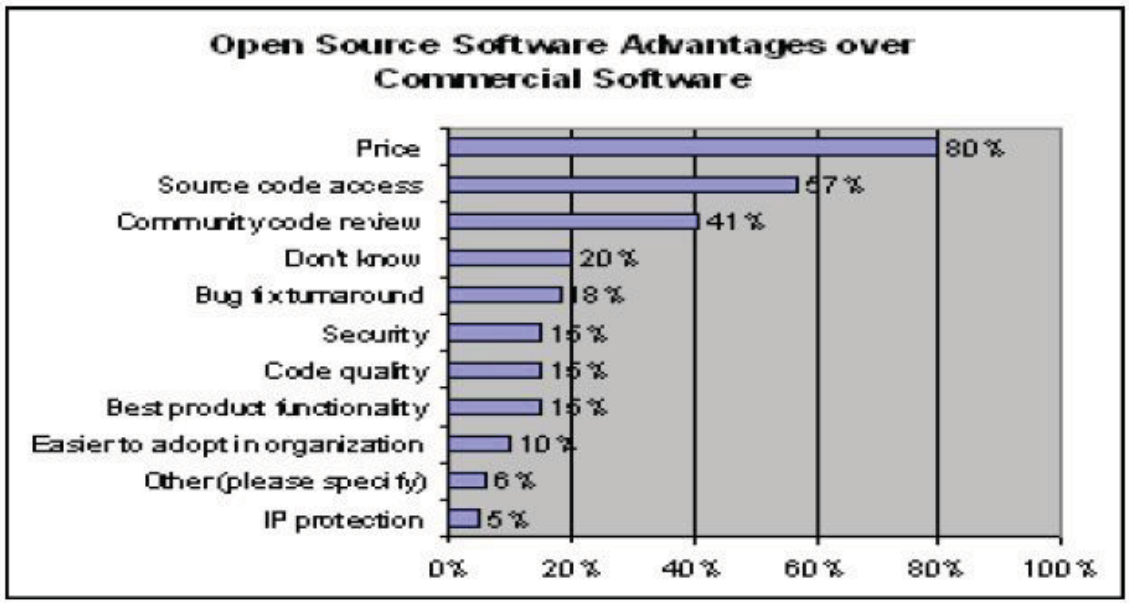


Fig. 2: Reasons for using Open Source products

The *UN,* in 2003:

> *calls on countries to adopt OSS to bridge the digital divide by lowering costs, increasing security, stimulating local economies, and avoiding proprietary lock-in as reasons for adopting OSS*

The trend towards using open source is clearly visible. Paul Daugherty, chief technology architect of Accenture explicitly expects

> *... this trend to develop as open source continues to evolve and to address even more business and critical functions.*

4.4 What Are Computers Used For?

Let's face it: these days most people use computers for handling e-mail and to do web-browsing. Additional use concentrates on office-like applications and maybe on the occasional artistic program (a drawing program, or a multi-media application). For these applications very good open source products are available. Thunderbird is a well-known open source program handling e-mail, Google Chrome's web browser already has pushed Internet Explorer from its dominant position in the browser world, and open source browsers as a group occupy approximately 75% of the browser market. Office-like applications like LibreOffice, a completely open source product, is available for more than 30 different languages, and on many operating systems, not just on Microsoft's systems.

What's left for closed source and proprietary software seems to be the personal computer market. This might be the only closed source stronghold that's left, existing mainly because of the impression instilled in the then home-computer users decades ago that software necessarily is expensive. However, by now it is clear to many countries, organizations, and computer users that there *is* a choice: open source operating systems and products allow us to steer clear of stifling contracts, to steer clear of expensive software, and to steer clear of low quality commercial or bootlegged products.

The choice you have is *not* a fairy tale: ***choose!***

References

Brokken, F.B., Kubat, K, Plomp, P.W. (eds), 1994: Proceedings to the First Dutch International Symposium on Linux, University of Groningen.

CDC Cyber: http://en.wikipedia.org/wiki/CDC_Cyber

Center for Strategic and International Studies (2010): Open Source Policies (document 100416_Open_Source_Policies.pdf)

Control Data Corporation: http://en.wikipedia.org/wiki/Control_Data_Corporation
ENIAC: http://en.wikipedia.org/wiki/ENIAC

Free Software Foundation: http://www.fsf.org

Gnu General Public License: http://www.gnu.org/copyleft/gpl.html

IBM: http://www.ibm.com

IBM PC: http://en.wikipedia.org/wiki/IBM_Personal_Computer

Linux (operating system): http://en.wikipedia.org/wiki/Linux

Microsoft Corporation: www.microsoft.com/

Open Source Initiative: http://www.opensource.org

Raymond, E.S. (2001): The Cathedral and the Bazaar, O'Reilly

Raymond, E.S.: http://www.catb.org/~esr/

Stallman, R.M.: http://en.wikipedia.org/wiki/Richard_Stallman

Stealth: http://stealth.sourceforge.net

Torvalds, Linus: http://en.wikipedia.org/wiki/Linus_Torvalds

5. Opensource Systems in City Management - From simple templates to business applications - how Open Source software improves public administration

Frank Siebert

Freies Office Deutschland e.V., Munich/Germany

Summary

In every municipality all over the world the tasks for managing city businesses are more or less the same: official documents have to be generated, approved and processed - the later by workflows, that vary from handing out a simple form to a multiple step processing including adding or editing data or database connection, concerning the complexity comparable with small business applications. The paper introduces the Wollmux, developed in Munich (Germany), a scalable and powerful opensource extension for template management and form generation. After an overview of the tools functionality its benefits are explained by means of example from the daily use.

Keywords

Wollmux, ODF, template management, form, business application

5.1 Introduction

In municipalities all over the world the tasks for managing city businesses are more or less the same: official documents have to be generated, approved and processed. The underlying workflow varies from handing out a simple form to a multiple step processing including adding or editing data or getting them from out of a database. So they often show a complexity comparable to business applications.

In the world of proprietary software there are a lots of available systems, which handle templates and document, connect documents with databases, provide text modules or help the user handling and completing forms. But unfortunately the implement of those systems is related with several sources for sometimes incalculable costs, especially for complex administrations, which are

- license fees, e.g. for the template system, office suite, interfaces
- customizing costs, e.g. adapting the system to the administrative needs
- training and instruction, e.g. courses and material
- effort for organizational changes due to functional restrictions

Especially in times of close budgets this monetary aspects weight double. Second there is a threatening vendor lock-in, when the dependency on a central product disallows a later migration – even when costs for support and upgrading are increasing more and more. To avoid the mentioned disadvantages the use of Opensource software seems to be an appropriate way – provided that adequate Opensource systems exist.

In 2005 the Munich Limux project [Limux - 2012] starts to migrate 80% of all administrative workstations from proprietary operating system to a free Debian based distribution, which was adapted to fit the existing processes and highest security demands. As a consequence of this step the office suite had to be migrated as well, because the existing and broadly spread proprietary office suite was not available for Opensource operating systems.

The decision was made to migrate to Openoffice.org. Even more important was the decision not only to migrate the huge amount of macros and forms found by

scanning workstations and server, but to review and consolidate each of them and transform them into a efficient administration-wide tool.

More than 30% of all found macros where of highly redundancy and dealt with different aspects of template and form automation. After an accurate review the former functions were summarized to the Wollmux, a flexible, scalable and powerful Extension to Openoffice.org for template management and form generation, written in Java. For very complex requirements a framework was designed to implement web applications for allocation of kindergarten or play-school [Trinuts - 2012].

Since 2009 the Wollmux is Opensource Software and available on www.wollmux.org under European Union Public License [EUPL – 1999]. The vivid community discusses on www.wollmux.net and provides an up to date wiki with lots of information for implementing and configuring the tool [Wollmux – 2012]

5.2 The challenge

The review of the discovered macros and a survey in the departments of the administration leads to a large number of requirements for the tool to be developed:

- Consistent corporate identity

  Each department used its own set of templates for internal and external communication. These templates should be harmonized concerning the use of fonts, styles, paragraphs and spacing, overall page layout and given information in header and footer.
- Consistent workflow

  Similar form should be processed in a similar way, independent of the department or location
- Reduction of complexity and effort

  Standard information like name, address and contact information of the user should always be up to date and filled in automatically.
- Reduction of errors

  The user should be guided wherever possible to avoid wrong information like date, file numbers or meta-data; form input should be format

and syntax checked to reduce mismatching information.

- Optimization of data integration

  To avoid redundant record-sets data should only be kept once for all purposes within the IT infrastructure; elements often used in different templates should only be stored as a single instance and then referenced by the templates.

- Convenient maintenance of templates and forms

  Even in complex structures the tool to be implemented should provide easy maintenance of all elements of the template and form structure.

5.3 The Wollmux – modules and functionality

So at the beginning there were a lot of requirements the tool named "Wollmux" had to deal with. It was clear that the tool should be implemented as an Openoffice.org extension, so that it could use all Openoffice.org libraries, functions and resources. The toll should be provided with and own menu, which gives access to the main functions and could be displayed within Openoffice.org.

Due to the limited resources of the project team function were grouped into modules, weighted and then implemented according the following order.

5.3.1 Module Template repository

Main module of Wollmux is the template handling. A Wollmux template is an Openoffice.org template file (*.OTT) which contain normal static text and *Wollmux fields.* Wollmux fields are an Openoffice.org bookmark containing a command string, that is automatically processed by the Wollmux, when the template is opened. The string follows the syntax

WM(**CMD** 'command' **PARAMTYPE** 'parameter')

The command for inserting an email address from the LDAP record-set of the user at the bookmark position is *WM(CMD 'insertValue' DB_SPALTE 'Mail').* The command for inserting another template at the position of the bookmark is e.g. *WM(CMD 'insertFrag' FRAG_ID 'Adresse_Angaben').*

The new inserted template can contain links to other templates as well. Because of the recursive structure of Wollmux templates central elements of all

templates, e.g. a city emblem, only have to exist as one template file for the whole administration. All other templates can make use of the emblem just by a Wollmux reference (example for standard letter head see figure 1 and figure 2).

There are a various commands which can be processed by Wollmux fields, e.g.

Also included in this module is the database connectivity, first of all already mentioned LDAP. By sending an LDAP requests the user information like phone number, mail address, location and all related information like public transportation to the bureau, or opening hours are received and used to fill the conform fields in the text fragments. Other existing database connectors are ODBC, JDBC and Openoffice.org Base.

5.3.2 Module Forms

Another very important module covers functions for generating and handling of forms. If a template contains Wollmux fields for form handling, there is an automatically displayed graphical user interface for all input and control fields (figure 4). This allows the user to fill in all field values not within the document, but convenient within the generated form.

The form handling covers some powerful function, e.g.

- gender function, which automatically adapts the document to a given gender value
- computing functions for processing numbers
- checking input values by format, range or content
- hide or show parts of the form GUI as a reaction to user input, e.g. if part of a form a irrelevant for a given case
- select address information from a database

5.3.3 Module Autotext

Wollmux provides a powerful mechanism for generating documents from a set of text modules. Wollmux text modules are inserted by the equivalent mechanism of a Wollumx field as described above. By inserting a special command combined with the text module name into a bookmark Wollmux is able to fill in

the referenced text. Again this mechanism is recursive, so a text module can contain reverences to other text modules as well. Text modules are located in a configuration file, but maybe also stored in a database.

This module is very useful for generating standardized documents like official notifications by citation of legal paragraphs or other given text modules.

5.3.4 Module Wollmux bar

The Wollmux bar is a small menu bar, which fixes itself at the upper border of the displayed desktop. Usually it is nearly hided in order not to block other programs menu bars to leave the whole display area to the user when not needed. By sliding over the small remaining part of the bar it swings open and shows completely configurable menu and button structure, which gives the user easy access to templates and functions (see figure 3).

The Wollmux bar is the only part of Wollmux functions the *normal* user gets to know. A standard configuration could contain e.g.

- a menu with access to the most often used templates for internal comments, external documents, memos or fax forms
- a set of subject-specific e.g. technical templates
- a menu with administrative forms for requesting resources, vacation form or business trip requesting
- buttons which give access to the most often used business application, like the well known windows start menu
- a powerful template search function

5.4 Examples

The following examples show the wide range of possible applications for the Wollmux with its different modules. First there is a brief description of the former process, then the situation with the use of Wollmux is described.

5.4.1 Simple: Application for leave

When people want to take days off for vacation usually the management want them to fill in an application for leave.

5.4.1.1 Without Wollmux template

Years ago there was a template for than, designed to match the corporate identity, including font type, spacing, emblem and given header and footer. The scan prior to the Wollmux implementation we found not less than 25 different forms for application for leave, none of them matching the original guidelines.

5.4.1.2 With Wollmux template

Every municipality employee has now access to a standard version of the form at the Wollmux bar with one click. When the template opens, the name and contact data of the employee as sender and the responsible manager as receiver are already inserted. All additional data are requested and simultaneous checked via the form GUI:

- start of absence (should be less or equal end of absence)
- end of absence (should be greater or equal start of absence)
- reason of absence (field should not be empty, required information)
- optional: information about availability in case of urgency

Duration of absence and number of workdays are calculated automatically. After finishing the form it can be printed (if necessary twice via a special Wollmux print button) or sent via email directly to the responsible manager.

5.4.2 Medium: IT Projects approval

IT projects, especially in the public sector, have to follow guidelines for design, standards, connectivity and security. To ensure the adequate participation of all related departments an approval form has to filled in by all participants in the initial phase of the project. Only after successful approval from the board the IT project has the clearance for going-ahead. Depending on the complexity of a single project the approval form covers 3 to 10 relevant pages.

5.4.2.1 Without Wollmux

Because of the missing possibility for hiding irrelevant parts of the document the approval covers always the whole questionnaire, even for simple projects. The

front page should display all relevant information. Therefor it has a complex layout which was accidentally destroyed almost every time when edited. The order of departments editing the document differs from project to project, and once inserted information could not be protected against unauthorized changes in the following process.

As a consequence of these difficulties the whole process took unreasonably long and documentation lack the desired standard.

5.4.2.2 With Wollmux template

Within the first step of the new Wollmux form the editor chooses the complexity of the project to be considered. Unnecessary parts of the original document are hidden, so the amount is reduced to relevant parts only. By using the form GUI user are not working within the form itself, so unintended changes to layout are effectively avoided.

Information given by one department can be set to read only for others, so only the editor himself can apply changes to the specific parts.

By using the Wollmux template for approval of IT projects the run to complete the document was reduced up-to 50%. Also the documentations follows strictly the given standards.

5.4.3 Complex: purchase form

This complex example would take several pages of explaining, so for now there is only a short description of how the Wollmux helped to reduce time and effort in the purchase bureau.

When a public institution resident in the EU is going to start a purchase process, there are a lot of different regularities to be considered. Depending on the type of product or service and the approximate overall prize the purchase form can last enfold a 3 pages form or 50 pages. And the difference between a local purchasing and a worldwide tender concerning the process is enormous: from 7 days for local purchasing up-to 200 days for a worldwide tender.

With the help of Wollmux the purchase bureau reduces their multitude of purchase forms to only one, which because of the flexible form GUI flexible enough to handle all kinds of tenders. At the same time the quality of the resulting form was increased significantly by using mandatory fields, format and syntax checking and automatic calculations with the form. The duration to finish the tender reduces up to 50%, and adaptations due to regal amendments is minimal.

5.5 Conclusion

Opensource software for public administration offers a modern and powerful alternative to proprietary systems, especially free office applications are in a straight line with administrative tasks processes.

The Openoffice.org extension Wollmux enhances the functions of the free office suit, so that not only creating and handling of templates are efficient and simple, but also complex business processes related to document handling are manageable.

Wollmux is designed to be highly configurable to fit for a wide range of IT infrastructure:

- local use only or enterprise wide use with integration of business databases
- simple templates or complex business processes
- national or localization for international use

In Munich and other German municipalities Wollmux has proven its ability to cover most of the task related to standardized written communication and should therefor be highly beneficial for other public administrations as well.

The examples in this paper are prepared with help of my colleagues G. Leichtl and M. Wedekind.

References

[EUPL - 1999]	http://joinup.ec.europa.eu/software/page/eupl
[Wollmux - 2012]	• http://www.wollmux.net/wiki/Hauptseite • Wollmux project site at OSOR.eu: http://forge.osor.eu/projects/wollmux/ • Wollmux repository: http://code.google.com/p/wollmux • Wollmux at Munich: http://www.muenchen.de/wollmux • Wollmux at Stadt Freiburg: http://wiki.stadt.freiburg.de/ooo/
[Limux - 2012]	http://www.muenchen.de/limux
[Trinuts - 2012]	http://www.kitavm.de

Appendix

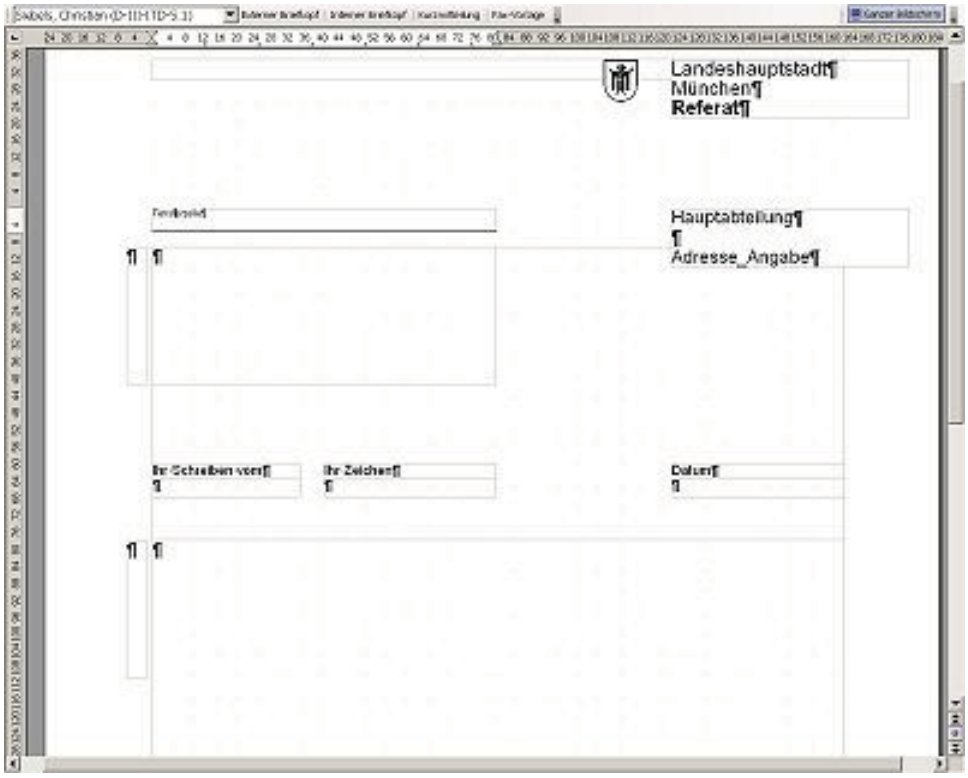


Fig. 1: Wollmux template BEFORE recursive processing of text fragments

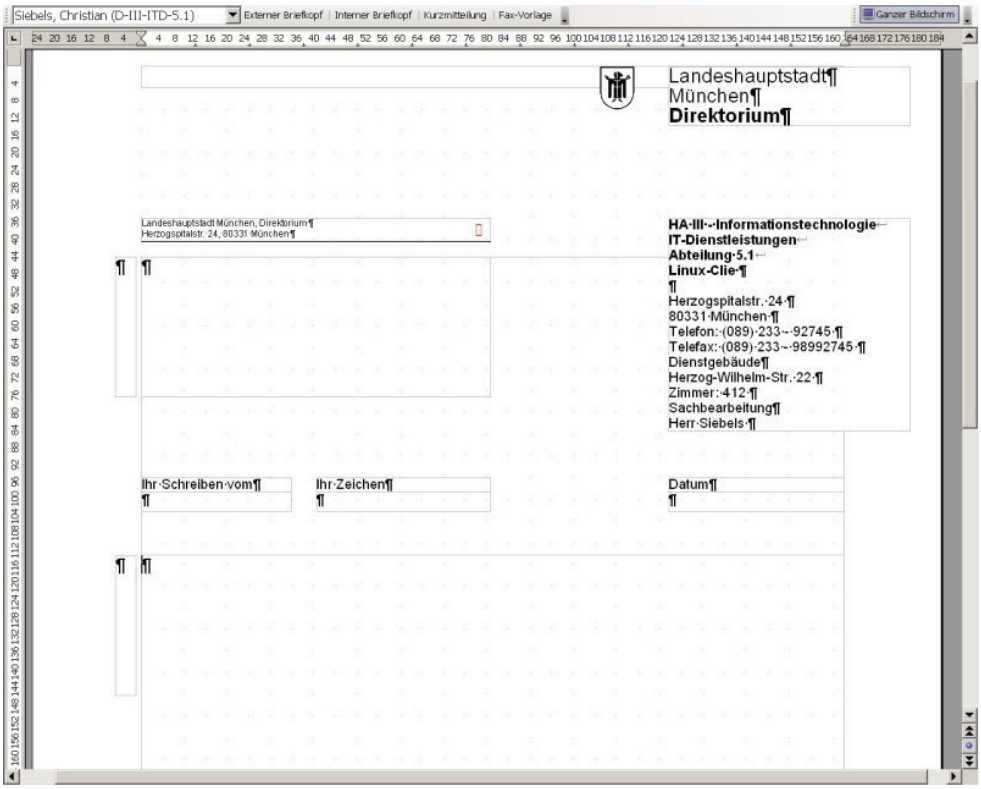


Fig. 2: Wollmux template AFTER recursive processing of text fragments

Fig. 3: Wollmux Bar - user interface for all central functions

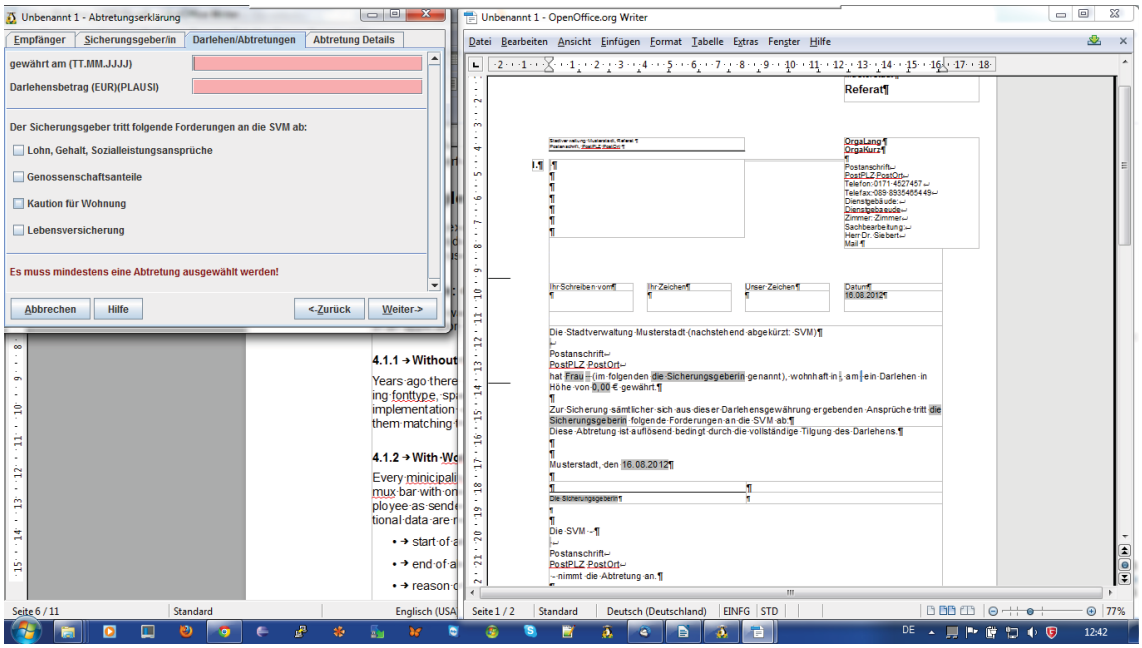


Fig. 4: Wollux form GUI with multiple tab layout

6. Organizational Structures within the Open Source Community

Matthias Stürmer

Ernst & Young Switzerland, Bern/Switzerland

	Single vendor open source project	Developer community	User community	Competence centers
Goals	Sell enterprise edition (dual licensing)	Follow business model and/or personal interest	Keep strategic control, share development costs, solve own problems	Promote open source in general, coordinate initiatives
Leaders	Initiating developing company	Developing companies and individual programmers	Software users	Board of directors of the organization
Scope	Global	Global	Global or regional	Regional or local
Examples	MySQL by Oracle, Openbravo ERP by Openbravo, Magnolia CMS by Magnolia	Linux Foundation, Apache Foundation, Mozilla Foundation, TYPO3 Association	Kuali Foundation for universities, PillarOne for insurance companies, GENIVI for car manufacturers	Open Source Business Alliance, Swiss Open Systems User Group, CENATIC, ADULLACT
Activities	Arrange partnerships, do marketing and PR activities, develop software in-house	Community building among developers, verify and integrate external contributions	Define common requirements, implement requirements in-house or contract external companies	Manage initiatives such as awards, directories, lobbying, information platforms, working groups etc.

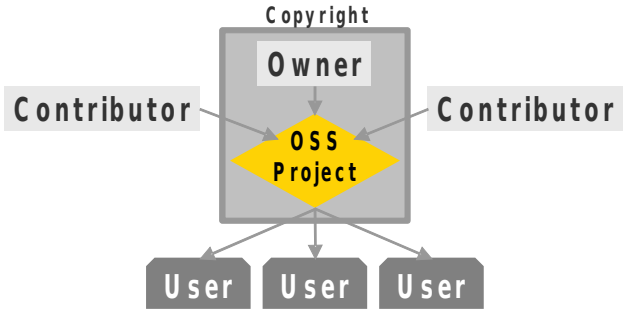
6.1 Introduction

Open source software is not only about programming code. There exist a vast amount of different organizational structures that facilitate the development and diffusion of open source software. In this article I explain the main types of organizations within the open source community. I distinguish between a) single vendor open source projects, d) development communities, c) user communi-

ties, and d) open source competence centers. These four types of organizations represent basic organizational patterns, variations and transitions between them are very common.

6.2 Single vendor open source projects

Often companies develop software in-house and eventually, for whatever reason, release the product below an open source license. They own the full copyright of the software thus are entitled to follow e.g. a dual-licensing model. This includes publishing the software under an open source license on the one side and selling the product through classical proprietary licenses on the other side. The company makes sure to remain the only copyright holder by letting external contributors sign commit or contributor license agreements. These contracts transfer the copyright of the contributed source code and additional assets to the software firm leading the open source project.



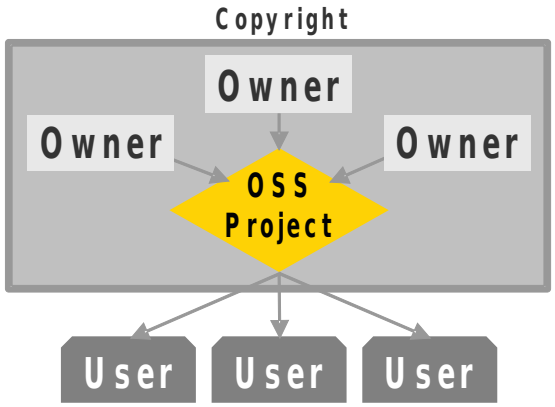
The primary goal of these open source projects are the sales of proprietary software licenses on a global scale. Gaining external contributors is challenging since giving up intellectual property rights to the leading firm provides little incentive to external software developers. Nevertheless reality has proven that even single vendor open source projects are able to build an open source community. I consider projects such as MySQL by Oracle, Openbravo ERP by Openbravo S.L., Magnolia CMS by Magnolia Inc., or Alfresco DMS by Alfresco Inc. as successful single vendor open source projects in terms of adoption within their respective market. Through company-lead developer conferences, partner programs, and other community building activities these companies managed to create a substantial amount of market share.

Sometimes critical people state that these single vendor open source projects are not real open source projects since there is a single company behind them controlling the entire community. As in proprietary software this may lead to vendor lock-in and thus an unsustainable product. I agree that users create dependencies with the leading company even if they don't decide to buy the enterprise version. However, since the identical source code is made available below an open source license there always exists the ability for unhappy users or developers to take the code, re-brand it and start a new open source project.

This so called forking of open source projects has happened many times in the past. E.g. currently there exist several forks of MySQL called MariaDB, OurDelta, or XtraDB. Also the open source ERP system Compiere lead by Compiere Inc. was forked in 2006 into the new open source project ADempiere. This proves that even such single vendor controlled open source projects are sustainable if a substantial part of the active developer community decides to move on with a new open source project.

6.3 Developer communities

Open source projects managed by developer communities represent the typical open source project. They were either initiated by an individual or by an organization such as a software company, a public administration, or a university. Through contributions from various sources the source code of the open source project is owned by multiple stakeholders.



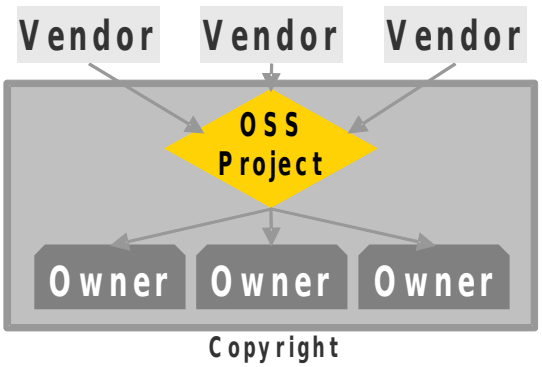
The contributors participate in the development process because of diverse motivations and follow different goals. Some programmers are involved because of ideological or other intrinsic motives. Some developers contribute because they need the software for themselves or because they want to learn something. Or they participate in the community because they are employed at an open source company or because they own an IT enterprise serving clients.

The open source project may be organized as a single community such as LibreOffice within The Document Foundation or the content management system TYPO3 within the TYPO3 Association. Or the project may be part of a greater open source association with various other open source projects governed by a foundation or other legal institution.

Examples are the Linux Foundation, the Apache Foundation, the Mozilla Foundation, or the Debian community. Such developer communities are organized in very different ways following individual rules and norms. They have in common that the control of the project is usually distributed among the main contributors (called meritocracy) and that the copyright of the software is shared among various stakeholders. Thus control is shared. There is no single commercial entity that may decide e.g. about the release process, feature road map, or the type of open source license.

6.4 User communities

User communities are similar to developer communities in the way that control is distributed among various stakeholders. However, in user communities the end user organizations of the software product own its copyright.



The software was either developed internally at a software user and then it was released as open source software. Or the software was created below an open source license as contract work by an external vendor. In either case the software users are the owners of the software product, the vendor provides services for the open source product but does not own the source code. Owning the copyright of the code allows users to define the software's type of open source license.

Users of open source products form user communities because they want to keep strategic control of their core systems and because they want to share the development and maintenance costs with others. Having a broad user basis of an open open source software product also may lead to external contributions such as documentation, translations, bug fixes, extensions, and eventually development of core components.

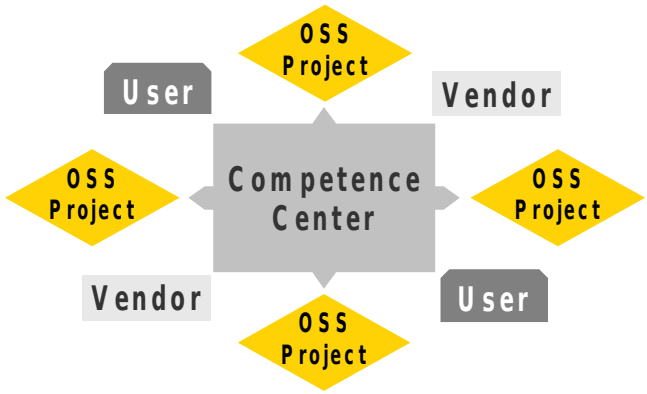
In their role as software users the main activity of the community is to define common requirements and implement them either by in-house development or by contracting external companies. An example of such a user community is the GENIVI Alliance, a group of car manufacturers (BMW, Renault, GM, Honda, Jaguar etc.) and suppliers (Continental, Bosch, Pioneer etc.) developing an open source in-vehicle infotainment. Or the re-insurance MunichRe and Allianz ART Insurance building the Enterprise Risk Management (ERM) suite PillarOne. Another example is the Kuali Foundation in the U.S. There are around 70 members of this user community, mostly large U.S. universities using the open source solutions for their campus management. In Switzerland there exists the OpenJustitia platform founded by the Federal Supreme Court of Switzerland to document and research previous court decisions, laws, and other legal databases.

6.5 Open source competence centers

As a fourth type of open source organizations there exist various forms of open source competence centers. They act as vendor- and product-neutral platforms with the goal to facilitate the use of open source software in public institutions, private companies, NGOs, etc. Open source competence centers close the gap

between software users, IT providers, and independent open source community members.

As activities an open source competence center typically organizes conferences and workshops, provides consulting services for administrations and businesses, and develops research studies and publications about open source. Furthermore open source competence centers often create directories of open source firms and credentials, build up and manage information platforms about



open source knowledge and experience, and the advocate the use and release of open source in government organizations and politics.

Open source competence center may also solve the challenge of collective action problems within open source communities: Often there exist common needs at a broad range of open source users for improvement of the software. However, since there is no single commercial entity that collects the little but numerous funding of the users, nothing changes. An open source competence center may coordinate development activities of certain user needs by defining the common requirements, pooling the resources, and procuring a vendor solution to the problem.

Successfully managed open source competence centers exist all over world [European Comission - joinup]. The initiating organization, the membership structure, the activities, size and funding etc. may vary. However, they all work with the common goal of improving the environment and general conditions to use and develop open source software. Such open source competence centers can be found for example in Switzerland (Swiss Open Systems User Group /ch/open) , in Germany (Open Source Business Alliance), in Spain (CENATIC),

in France (Adullact), in UK (OSS Watch), in Norway (Friprog), and in Europe in general (Open Forum Europe).

References

European Commission – joinup – Open-Source Software, http://joinup.ec.europa.eu/software/page/oss-competence-centres

7. Open Source in Mixed Environments - Using Open Source to improve security and throughput

Frank B. Brokken

University of Groningen, Groningen/Netherlands

7.1 Introduction

Organizations usually harbor a variety of computer systems. Network and main server computers usually run variants of the Unix operating system, and the work stations of employees are often MS-Windows or Linux systems. In these mixed environments maintenance must pay attention to the specific requirements of these different types of computer systems.

While each operating system has its own specific needs and requirements, an organization must make sure that the security of all of its computer systems is not easily compromised, as this endangers the organization's continuity. In this paper some of the highlights of organizing computer security in a mixed computer environment are covered.

First some general concerns relating to the security of computer systems are discussed. Following this a generic network setup is covered. The basic terminology and organization of firewalls is briefly outlined, and an example is given illustrating how easily even a severely restricting firewall can be bypassed. Consequently, an organization should not place all its trust in its firewall, assuming that this will offer sufficient protection against intruders. Several open source

products are available implementing additional protective measures, and some of these open source products are mentioned in this paper. These days organizations often require more bandwidth than is immediately available. Organizations can use bandwidth management to distribute the available bandwidth over its various departments and users. Again, open source products are available for implementing bandwidth management.

This paper is accompanied by an extensive slide presentation. The reader is occasionally referred to these slides for additional illustration and documentation.

7.2 General Concerns

Quoting Richard Clarke, former special advisor to the US president on Cyber Security:

> *If you spend more on coffee than on IT security, then you will be hacked. What's more, you deserve to be hacked.*

Taking his quote seriously, then many ordinary computer users deserve to be hacked. Usually computer users pay little attention to the security of either their computer systems or the actions they perform with their computers. Unfortunately, this often also holds true for the technicians who are responsible for the proper maintenance of their computer systems. Even worse, it holds true for many organizations as well.

Organizations, at the very least, should have published an *Acceptable Use Policy* (AUP) describing the proper use of its information technology (IT) facilities. Many organizations, like the University of Groningen, have published AUPs and have informed their employees (and, for educational organizations: their students) that they should not violate the organization's AUP.

Although AUPs offers foundations for enforcing proper IT use, users can easily compromise the security of their computer systems. In many situations where information is exchanged between computers this information is transmitted as clear text, allowing anybody along the route between the sending and receiving computer to read all of the exchanged information. Some of these commonly used clear-text protocols are e-mail (*smtp*), the *http* protocol used by the world

wide web, the *telnet* protocol, the *smb* file sharing protocol and the *rpc* remote procedure call protocol.

It is illustrative and often amazing to watch the route information is taking when it is sent from one computer to another. The trace-route utility can be used to determine this route. Figure 1 shows the route packets take from my computer in the Netherlands to a computer in Afghanistan.

```
traceroute to 58.147.128.10

 1  129.125.3.252  (NL)
 2  145.145.2.121  (NL)
 3  145.145.80.77  (NL)
 4  109.105.98.33  (SE)
 5  109.105.97.66  (SE)
 6  64.57.21.53    (US)
 7  206.223.115.48 (US)
 8  63.218.211.86  (US)
 9  210.5.200.9    (HK)
10  210.5.192.19   (HK)
11  210.5.197.2    (HK)
12  58.147.128.10  (AF)
```

Fig. 1: Traceroute from 129.125.3.163to 58.147.128.10

The route taken for this connection is indeed remarkable. Instead of going directly towards its destination (one would assume that, e.g., Turkey would be one of the visited countries) the packets almost circle the world, allowing computers in the USA and China (Hong Kong) to pick up and read the complete communication. Picking up and interpreting information that is sent using clear text protocols is easy. *Tcpdump* is a network packet analyzer allowing computers along the route connecting two computers to pick up all the information that is exchanged between these computers.

Another useful tool is *Wireshark*, allowing us to analyze the sniffed packets, turning them in human readable form. Users exchanging confidential information via e-mail or through http web-forms may find, to their dismay, that the confidentiality of the exchanged information was already compromised before the information even reached its intended destination. It gets even worse: the *man-in-the-middle* might even have changed the information while it was passing through the bad guy's computer.

Unfortunately, system administrators, because of the additional privileges they have, may also cause the integrity of the computer systems they maintain to be compromised. A system administrator should be able to answer the following questions in a satisfactory way:

- What programs are available on your computer?
- What is your upgrade policy?
- Are all the computer's accounts actively being used?
- What services does the computer offer to the outside world?
- How to you inspect the computer's log files?

System administrators unable to answer these questions may find that intruders already have gained access to their systems using old, forgotten accounts, using well-known bugs in old software, or using vulnerabilities in the software listening to ports the systems are offering to the outside world. Such system administrators should be replaced by more competent administrators.

Incidentally, open source programs are available for answering the above questions, and there is really no excuse for any system administrator for not being able to answer the above questions.

7.3 Network Organization

The complexity of networks used in organizations greatly varies. Figure 2 shows a generic network organization that can easily be expanded, but that also shows many of the essential characteristics of a network. Information from and to the Internet reaches the organization via a firewall, where the information is routed to (or from) different segments of the network. Departments or units use their own separated section of the network, which is independent of the public services the organization offers.

Figure 2 merely shows a single internal network, but in reality multiple internal networks may exist, each servicing a separate section within the organization. All public services (like the organization's e-mail server, the organization's website, etc.) are separated from the internal net, and are placed inside a so-called *demilitarized zone* (DMZ) to prevent intruders from reaching the internal net-

work. Network and system administrators should assume the servers in the DMZ are in fact compromised and should approach its computers with the utmost caution.

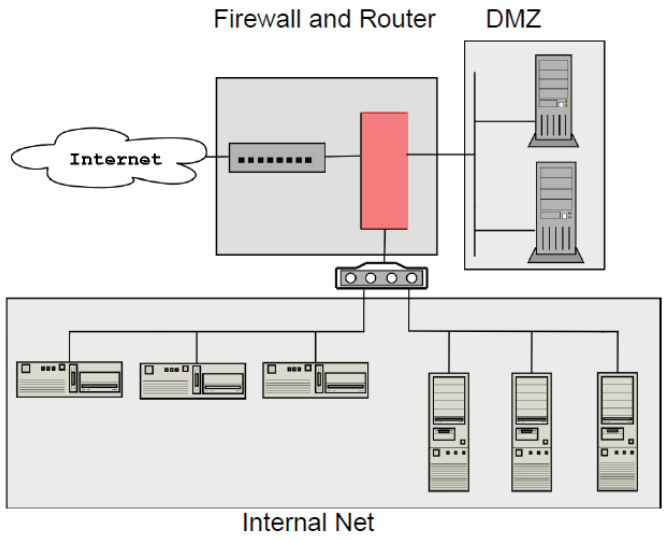


Fig. 2: Generic NetworkOrganization

The firewall/router part of the network organization shown in figure 2 must ensure that each of the sections of the network can only perform their assigned tasks. For example, information from/to the DMZ should have the following characteristics:

- Incoming traffic is only allowed to services that were defined in the DMZ.
- Outbound traffic can only be part of established connections
- New connections cannot originate from the DMZ
- Information leaving the DMZ can only use the addresses of the host that are defined in the DMZ (simple plain spoofing protection)
- No information is forwarded to other segments of the network

The other segments of the network show comparable characteristics, which are summarized in the accompanying slides.

A generic network setup like the one presented here can easily be organized and monitored using open source programs. The implementation of

the Linux firewall is provided by *Netfilter*. The organization of a Linux firewall is usually controlled by the *iptables* program.

When defining the firewall rules, *forwarding* refers to the information that merely passes through the computer. E.g., when information reaches the computer running the web-server (inside the DMZ) and that computer passes the information on to one of the workstations in the internal network, then the computer running the web-server is *forwarding* that information. This should not be possible. Instead, information reaching the computer running the web-server must stay inside that computer. This is called the firewall's *input* chain. There also exists an *output* chain: information leaving the computer after it has been processed by one of its local processes. All these chains, as well as the additional processing that is required when a network uses *Network Address Translation* (NAT) are illustrated in the accompanying slides.

7.4 A Simple Vulnerability

It is not sensible merely to rely on a firewall for securing your network's computers. In general a *multi-layered defense*, using different methods assuring the integrity of the computers, should be deployed. No single method offers perfect security, and even a very strictly configured firewall can be circumvented by a clever intruder. The accompanying slides illustrate how an intruder may gain access to a sensitive service, which should be available only to employees within the organization using a so-called *ssh-tunnel*.

The tunnel can be established from the outside world to a computer within the organization. This allows the intruder to gain access to the sensitive service, even though the organization's firewall does *not* allow incoming connections at all. Tunneling is a well-known technique, which has many useful applications (like setting up *virtual private networks* (VPN)), but an SSH-tunnel can also be used to gain access to an organization's network illegally.

To set up an SSH-tunnel the intruder must have been able to execute a command at a workstation within the organization. In practice this is easily accomplished. The *Common Vulnerabilities and Exposures* (CVE) database shows a large number of exploits against MS-Windows systems, allowing an intruder to

execute any command at the targeted system. Alternatively, the required command may be executed by a disgruntled employee, or the intruder can use *Social Engineering techniques* (cf. Hadnagy, 2011) to goad employees into executing the required command.

7.5 Vulnerability and Integrity Monitoring

Considering the number of exploits stored at the CVE data base, and considering how many people out there might be interested in illegally gaining access to computers, and considering that system administrators may not always be able to closely monitor the integrity of their systems, we should assume that, without evidence to the contrary, that at least some of the computers in an organization have been compromised.

Fortunately, many Open Source products are available to verify the integrity of computer systems. The *Open Source Vulnerability Scanner* (OpenVAS) offers a very large number of exploits that can be launched against computer systems. In many cases these exploits are operating system specific, and OpenVAS allows you to select those exploits that are operating system specific. Having completed its vulnerability scan, OpenVAS can produce reports in various forms and formats.

The reports provide detailed information about all the items OpenVAS noticed, distinguishing *notices, warnings* and *holes*. Notices merely report features of the inspected system, like its type of operating system or the version and type of its web server software. This is an important awareness tool: you might not be aware that some of these reported services are actually provided. Warnings and holes require more specific attention, and OpenVAS not merely mentions these warnings and holes, but in addition provides information about how to prevent the warnings from being raised or how to plug the encountered holes in the computer's security.

Figure 3 shows a bar-diagram type of summary, produced by OpenVAS after scanning a computer. In this particular case OpenVAS merely reported notices, and no warnings or holes.

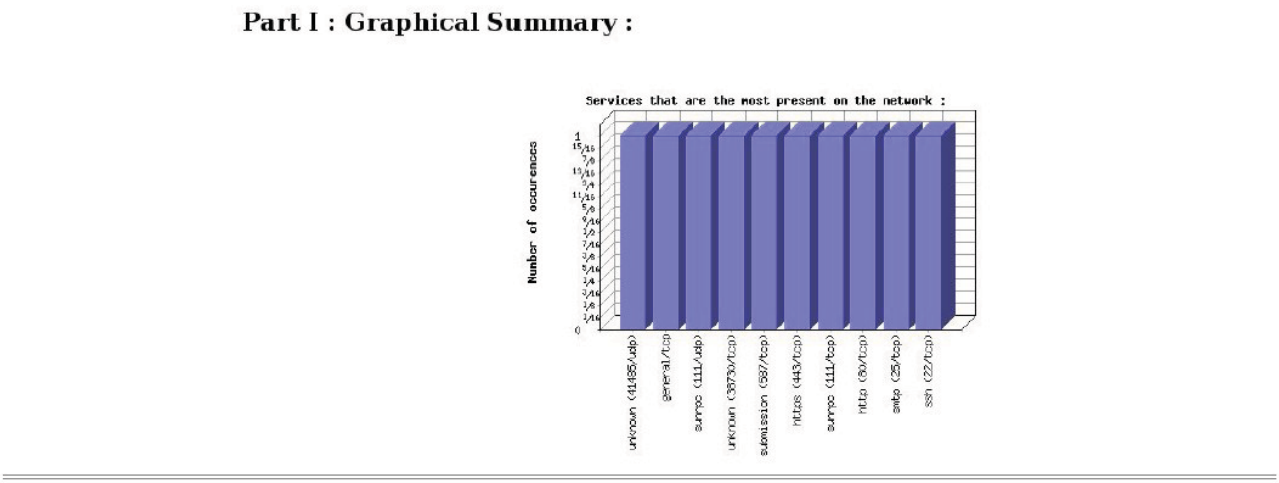


Fig. 3: Example Summary Report as produced by OpenVAS

Eventually, a computer may become compromised. There may have been a zero-day exploit against a system, allowing an intruder to gain access to one or more computers within an organization. Then what?

In all situations where an intruder gains access to a computer system the intruder will at least modify one or more of the essential characteristics of the compromised system. Programs may have been modified allowing the intruder future entry through a *back-door*, or the computer system may be abused by the intruder to allow him/her to store illegal or copyrighted materials within the organization's computers.

These situations are covered by *Integrity Scanners*, like the Open Source File Integrity Monitoring program *Stealth*. File Integrity Monitors typically compare cryptographic hash values (like the values computed by the *sha1sum* program) of files stored inside a computer with previously computed hash values. Unexpected discrepancies may indicate that files have been modified, which should be investigated by the system manager who is responsible for the affected computer system.

7.6 Bandwidth Management

Frequently, organizations do not have enough bandwidth available. Internet use shows an ever-increasing curve, and an organization's bandwidth may not be

enough to unrestrictedly service all its employees. If nothing is done about this then congestions will be the result, in which some user or section of an organization's network could monopolize the complete bandwidth.

In cases like these, the use of *bandwidth management* is indicated. Bandwidth management is an extensive and complex topic, and here, as well as in the accompanying slides, this topic can only be highlighted.

Although the topic itself is complex and extensive, its implementation tools are not. Bandwidth management can be realized using the well-known *iptables* program, in combination with another open source tool: *tc* (for *traffic control*). The *Linux Advanced Routing and Traffic Control* (LARTC) website is a good starting point when preparing for bandwidth management.

Bandwidth management is normally implemented on a outbound network interface. Several forms of bandwidth management can be used, each serving a particular situation. Combinations of these various forms of bandwidth management are also possible. In this paper some of the more frequently occurring situations are briefly mentioned. The reader is referred to the accompanying slides or to the documentation at the LARTC site for implementation details.

Traffic may be subdivided into certain types, e.g., interactive and non-interactive traffic. Consider this situation: if many computers are busy visiting website or sending e-mail then their use of the available bandwidth may easily suppress or slow down interactive sessions (e.g., an *ssh*-session). Interactive sessions usually require very little bandwidth, but the human being at one end of the connection expects to be able to enter his/her commands without interruptions. In situations like these the *pfifo_fast queueing discipline* may be used.

Using *TCP*'s *type of service* (TOS) field interactive sessions may easily be given the highest possible priority. This way interactive sessions no longer suffer from massive down- or uploading, while the interactive traffic, on the computer's time scale, occur so infrequently that it doesn't noticeably slow down other types of traffic.

Another useful form of bandwidth management is the *Stochastic Fairness Queueing* (SFQ) discipline. Here a round robin approach is used to allow each host to use of the bandwidth for a short interval of time. No host is able to mo-

nopolize the connection, but instead each host is given a slice of time to use the connection.

In complex organizations different sections may require different amounts of bandwidth. In the DMZ network organization described earlier the internal network may require a larger bandwidth than the DMZ itself. When there are multiple internal networks, bigger internal subnets may require a larger bandwidth than smaller internal subnets. This situation, where the available bandwidth must be distributed (possibly using unequal proportions) over the various segments of the network, can be implemented using the *Hierarchical Token Bucket* (HTB) queueing discipline. The HTB queueing discipline also allows network segments to `borrow' bandwidth from other segments, if these other segments temporarily do not require the use of all of the bandwidth that has been allotted to them.

7.7 Conclusion

In many organizations a mixed IT environment is encountered. Maintenance of these mixed IT environments is a complex task, which often transcends the level of the operating systems that are being used. Irrespective of the operating systems that are being used, the organization should ensure the security of its computer systems and of its network organization. This not only involves deploying technical means, but also user education, as well as publishing and enforcing an Acceptable Use Policy.

Many open source tools are available for maintaining and monitoring the network organization and the integrity of the individual computer systems. Programs like *iptables, OpenVAS* and *Stealth* can be used to improve and monitor the organization's IT infrastructure.

Finally, open source tools are available for distributing the available bandwidth over different segments of an organization's network or for assigning different priorities to different types of services.

Many more open source programs than the ones mentioned in this paper can be deployed in mixed IT environments. In this paper I could only show a small tip of the proverbial iceberg, and the reader, using Google or the overviews pro-

vided by the various Linux distributions, will usually be able find an open source tool that perfectly fits his/her maintenance task at hand.

References

Backdoor Attack: http://en.wikipedia.org/wiki/Backdoor_(computing)

Bandwidth Management: cf: Linux Advanced Routing and Traffic Control.

Common Vulnerabilities and Exposures Data Base: http://cve.mitre.org

File Integrity Monitoring: http://en.wikipedia.org/wiki/File_integrity_monitoring#Applications

Hadnagy, C. (2011), Social Engineering: The Art of Human Hacking, Wiley.

Http: the World-Wide-Web's hypertext transfer protocol: (http://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol)

Iptables: the Linux firewall administrative tool (http://en.wikipedia.org/wiki/Iptables)

Linux (operating system): http://en.wikipedia.org/wiki/Linux

Linux Advanced Routing and Traffic Control: http://www.lartc.org. See also http://www.funtoo.org/wiki/Traffic_Control.

Multi Layered Security Plan: http://cybercoyote.org/security/multi-layer.htm

Natlog: A Network Address Translation Logging Tool (http://natlog.sourceforge.net)

Netfilter: the Linux packet filtering firewall (http://www.netfilter.org)

Network Address Translation: http://en.wikipedia.org/wiki/Network_address_translation

OpenVAS: Open Source Vulnerability Scanner (http://www.openvas.org/)

OpenVPN: Open Source implementation of a Virtual Private Network (http://openvpn.net/)

RPC: the Remote Procedure Call protocol (http://en.wikipedia.org/wiki/Remote_procedure_call)

Sha1sum: http://en.wikipedia.org/wiki/Sha1sum

SMTP: the Simple Mail Transfer Protocol
(http://en.wikipedia.org/wiki/Simple_Mail_Transfer_Protocol)

SMB: the Server Message Block file sharing protocol
(http://en.wikipedia.org/wiki/Server_Message_Block)

Stealth: An Open Source File Integrity Monitoring Program
(http://stealth.sourceforge.net)

Tcpdump: a packet analyzer (http://en.wikipedia.org/wiki/Tcpdump)

TCP/IP Guide:
http://www.tcpipguide.com/free/t_IPDatagramGeneralFormat.htm

Telnet: the traditional command-line interface protocol
(http://en.wikipedia.org/wiki/Telnet)

Traceroute: a tool displaying the route packets take across the Internet
(http://en.wikipedia.org/wiki/Traceroute)

Tunneling Protocol: http://en.wikipedia.org/wiki/Tunneling_protocol

University of Groningen: Acceptable Use Policy
(http://www.rug.nl/cit/security/aup/)

Virtual Private Network (VPN):
http://en.wikipedia.org/wiki/Virtual_private_network

Wireshark: a network protocol analyzer (http://www.wireshark.org/)

8. TechSession: Web-Centric Application Development - an OpenSource framework for business applications in public administration

Frank Siebert

Freies Office Deutschland e.V., Munich/Germany

Summary

Web-centric application development is a powerful method to supply complex functionality combined with high efficiency and best performance even with older hardware. But sometimes high level skills are required to implement web applications, sometimes restrictions concerning e.g. security forbids the usage in public administration. This paper introduces the web application framework Ordo, based on a concept which was developed in Munich (Germany) for rapid development of web applications. Ordo is perfect for missing or unspecific user requirements and leads to application standardized in design and usage. Therefor Ordo reduces the effort for maintenance and support as well as it serves for web applications easy to handle and with low educational costs – optimal characteristics for the use in public administration.

Keywords

Ordo, business application, web-application, rapid development

8.1 Introduction

In 2005 the Munich Limux project [Limux - 2012] starts to migrate 80% of all administrative workstations from proprietary operating system to a free Debian based distribution, which was adapted to fit the existing processes and highest security demands. Knowing that this means, that hundreds of business applications will not longer run – at least in the existing MS Windows Version – several steps were performed for each of them:

1. check if there is already an alternative included in the standard debian client
2. check if there is already an adequate alternative existing in another department of the administration
3. analyze the market for platform independent OpenSource software alternatives or at least proprietary systems
4. build an own platform independent alternative.

In municipalities all over the world the tasks for managing city businesses are more or less the same – according to this fact there are a lot of commercial systems for all parts and demands of a public administration. But choosing proprietary software often means to accept some unattractive aspects regarding effort and costs, as for example

- costs of acquisition and license fees
- customizing costs, as adapting the system to the administrative needs
- training and instruction, as courses and material
- effort for organizational changes due to functional restrictions
- effort for adapting the proprietary system for a possible change to a new operating system

So if there was not already an alternative business application included in the standard repository, the best alternative concerning costs and function often was the implementation of an own system, perfectly customized to the specific user requirements and tasks.

The situation was quite comparable to the result of document and template review, which lead to the development of Wollmux [Wollmux – 2012]., The Open-

Source template system Wollmux was especially designed to fit most of the administration needs for comfortable template and form handling. Encouraged by the success of Wollmux, a project was set up to design a framework for business applications, running in the administrations intranet.

Following the concept of generic web-centric application development the resulting framework is nowadays responsible for more than 150 applications, ranging from highly specific systems for city planning (5 users) to broadly used index for contact and organizational data of the whole staff (> 18.000 user). While the use of this framework is still restricted, an OpenSource alternative is developed [Trinuts – 2012].

8.2 The challenge

The requirements for applying the wanted framework in contrast to other tools, e.g. Wollmux, are simple and well defined. A web application should be implemented in case of

- multi user operation
- multi site access
- underlying database structures
- special function like email or PDF generation of documents

The review of the business application to be implemented by the intended framework leads to a number of requirements:

- Consistent handling

  The intended framework should reduce the variety of user interfaces for business application. Applications should follow a standardized style guide and similar functions should be handled in the similar way to reduced costs for training and optimize the benefit for the user.
- Optimal integration in existing infrastructure

  The framework should integrate optimal into the existing infrastructure, as LDAP (user and department information) and database systems (application data like GIS, mainframe databases for administration office). Data import and export of given data sources should be indepen-

dent of the type of database. The output of user documents should be in open format, e.g. Open Document Format (ODF), so further processing via an OpenSource office suite (Openoffice.org, Libreoffice) is possible.

- Missing or incomplete user requirements

  In most of the cases a detailed requirement specification is not available. The specific need is more or less a general need description of the people, who want to use the application, in best cases a operation breakdown. So the framework should support a repetitive process including rapid prototyping and easily adaption to changing requirements during implementation.

- Convenient maintenance

  Because of the sheer number of new applications and the limited capacity of technical staff the framework should offer convenient maintenance features, as simultaneous update of more than one application or the reused of function in other application.

- OpenSource and open standards

  The tools for creating the intended framework should exclusively be available as free and OpenSource software themselves. Due to the idea of OpenSource software this requirement should avoid possible restrictions (for example only limited functionality of proprietary libraries, unknown security leaks).

8.3 The framework elements

As already mentioned the concept of the framework was developed for the municipality of Munich, Germany. As the Munich framework is still not available as OpenSource, this section describes the Ordo framework. Actually the ancestor of Ordo was the starting point for the developments made in Munich. Differences between the Munich framework and the newer Ordo framework are shown in fig. 3 and fig. 4. Ordo will be available under European Public License (EUPL) [EUPL – 2007] until end 2012.

8.3.1 Underlying tools

The most important requirement for the intended framework was the use of OpenSource tools for its implementation. Fortunately there is a wide range of OpenSource programming languages, application server and databases since long. They are well documented, tested and supported by a vivid community.

8.3.1.1 Programming language

As the programming language for the implementation of the intended framework Python was chosen. Python is an universal, object oriented OpenSource language, which is used for complex applications as well as small scripting tasks. Its quite readable and runs under Windows, Mac and Linux and is also available for mainframes and supercomputers.

8.3.1.2 Web server

The most common web server is Apache. Its version 1.0 was released in 1995, and the fact that there are important security updates for 1.0 indicates that it is still in use! It is available for nearly all platforms and due to its popularity there is a very active community.

8.3.1.3 Database

The most popular OpenSource database system is MySQL. Nonetheless PostgreSQL was chosen because of its similarity to Oracle, the most often used proprietary database system. So it should be possible to migrate to Oracle - if needed at all – with only a minimum of effort.

8.3.1.4 Browser and Layout

The independence of browser types and versions was crucial, because especially in public administration often older versions of browser are and still stay in use. Despite of browser depending differences in style and function handling a similar layout of the web application was demanded. To ensure the best possible result the free framework for cascading style-sheets (CSS) YAML (Yet Another Multicolumn Layout, [YAML – 2012]) was used.

8.3.2 Module structure

The structure of the framework was kept smooth and simple to reduce maintenance effort (see Fig.1).

8.3.2.1 Web/Apache modules

There are two modules working as interface to the Apache web-server. All responses and requests were handled here.

8.3.2.2 General Modules

There is a group of general modules, which provides sets of services to the core modules. This are

- **EnvironmentVar:** environment, application and dynamic session variables to store the actual state per user
- **Database:** database connectivity, handling of SQL statements for manipulating data or requesting parameters
- **Exception:** error handling for incorrect response of modules, functions or database
- **Utils:**utility functions like parsing, user permission for application elements or generating standard HTML output strings
- **Print:** handling of print commands, generating of documents and their delivery to the web-server for download

8.3.2.3 Ordo Main Modules

The Main modules are connected to each other in a strictly hierarchical manner and are shown in fig. 1.

1. **Frontdesk**

   A Request from the WSGI module is handled by the module *Frontdesk*. The correct user login is checked, then the correctness of the delivered request string. In case of a new login the session is prepared, in case of logout the session is killed. The type of item (*page call, button pressed*) is determined.

2. **Itemhandler**

   Depending on the given item session variables are set to control the further processing.

3. **Items**

   The module *Items* processes the different item types like *page call, button pressed, login/logout* or *download requested*. If item type was *button pressed*, the processing is routed to module *Events*, else to module *Generate_HTML*.

4. **Events**

   This module handles the different event types like save, new, search, print or others. Depending on the type of event session variables have to be set for the further processing and HTML code generation.

5. **Generate_HTML**

   This module takes generation of all HTML output and returns it to preceding modules. Due to the tree structure of pages the calling path within the module is always the same. First the *generatePage* function is called, which delivers the main elements of the later HTML page like doc type, DTD link and debug infos. It then calls *generatePageHeader*, *generatePageElements* and *generateFooter*.

   generatePageElements includes javascript libraries and calls *generateMenu* and *generateMain*. Here the *generateSearchResult* function is called and 1–n times the *generateForm* function. A form can include1-n buttonrows, 1-n fieldsets, 1-n tables. Also each of the former can include 1-n form elements, which could be of following type: buttonrow, button, chart, checkbox, countlink, fieldset, hidden, image, input, javascript, label, link, map, password, radio, select, table, textarea, treeview. Each form element has its own generate-function, which adds a piece of code to the HTML page.

8.3.3 Definition of application elements

Each page, form and form element is stored as a record-set in the core table of the framework. This table consists of 61 columns, which define the following properties:

- Application, id, mask number, mode type, parent element id
- form element type, position in the form, element dimension and css class
- related database field, check pattern for input, javascript
- SQL to be processed
- next mask number, next mode,
- label, tooltipp, info and help text in 5 languages
- access authorization for 6 different user profiles
- conditional appearance and accessibility as result of e.g. SQL requests

In this manner all elements are stored in the core table. For each request the specific records are selected and and proceeded as shown in Fig. 2.

8.3.4 Main features

Structure and functions described in this chapter lead to some features, that made this framework cope with the difficult requirements listed in .

- The Ordo framework allows the flexible allocation of business logic between the application definition table and the application database. Record-set order, manipulation of record-sets, actions while entering or leaving a page or accessibility can be coded in tables and views or in the core table. So depending on the skills of the team the application can be implemented database centered or core table centered.
- The effect of changes in the core table are visible with the next request for corresponding page. If there is no underlying table for displaying data, form elements are displayed in demo mode. That offers the possibility to generate workflows and forms before the database is available. So especially if there is no or only poor user requirement specification Ordo framework can be used as a rapid prototyping tool. In most the examples described in this possibility was crucial for the success of the project.
- All application can either run with individual modules or with the same

main modules. The second means, that Ordo framework is more or less an application server. Functional extensions triggered by a new project are instantly available for other applications as well

- Due to the simple HTML code of the output application a good performance is achieved even with low or poor connections. This allows to integrate external staff on any locations, regardless of the type of connections.

8.4 Examples

By now there are more than 150 different applications used by approximately 20.000 user in their day-to-day business. In this sections only a few applications are described which are special because of their large number of users, their complexity or their enormous advantages for the public administration.

8.4.1 Interaction between departments

8.4.1.1 Telephone directory (~18.000 user)

The directory contains contact data and information concerning the organization of all public employees. All data are stored in the central LDAP database. Phone and contact information can be found via search forms, organizational information like department, office location, coworkers or supervisor are accessible with one click via link.

8.4.1.2 Location plan (50 user)

Building applications have to be checked by different departments. The review has to follow a distinct order, so the department an the particular office in charge changes several times. The application controls the workflow by showing or hiding the respective form fields.

8.4.2 Department of Social Welfare

8.4.2.1 Tracking of social services for risk groups (700 user)

If social services get a hint at difficult private situations in a family, they try to act as a coach and take care of the involved children. All actions then have to be

well documented to guarantee best possible reaction. All contacts are filled into a database and are reviewed by coworkers. Due to the sensitivity of the data the application has got a detailed authorization and access management.

8.4.2.2 Allocation of children to play-school (350 user)

This application helps the staff of play-schools to cope with thousands of applications for a place within the municipality(fig. 3). All data are stored in a central database, which later provides distinct suggestions for formation of each new group of children. Based on this concept Akita (fig. 4) was developed, and today other public administration are already using the application [Trinuts – 2012].

8.4.3 Department of Building and Construction

8.4.3.1 Tracking of damages of public infrastructure (100 user)

Public infrastructure has to be examined for damages. The division responsible organizes inspection tours trough the whole city. After each tour the inspectors enter their findings into the damage tracking database. Even private people can fill in a standard paper form to report damages, which is transformed into the database, so there is a detailed overview of infrastructure conditions for a given moment.

8.4.3.2 Simplified single tender action (15 user)

This application generates single tender actions for example standardized repairing of infrastructure elements like streets, traffic lights, public illumination, . Text modules are combined depending on the kind of repairing and a standard order for repairing. An automatically database interface to accounting systems is implemented.

8.4.4 Other departments

All department use several framework applications, here are only some more examples:

8.4.4.1 Timetable for mayor and deputy mayors (4 user)

Even the mayor and deputy mayors of the municipality of Munich are using a web application, because the amount of regular appointments and requests for joining public events demand more functionality than the standard calendar offers.

8.4.4.2 Board decision template (70 user)

Due to legal aspects board decisions have to follow some very strict standards. To make sure these standards are followed exactly, all information are sampled by the application and stored in a database for documentation. Depending on the subject of the board decision to be made a more or less complex document is automatically generated. Formal errors are avoided and board decision documents are stored for later auditing.

8.4.4.3 Delivery index (10 user)

The department of Water and urban drainage allows private disposal of sewage water and sludge. The delivery to the wastewater treatment plan has to be tracked for charging the delivering company for disposal efforts. The application runs at two sites, where the gate keepers do the weighting of in- and outgoing trucks. The administration clerk addresses these data for generating the corresponding bill.

8.4.4.4 Rent index (20 user)

In the office of Statistical Information data are collected of rent levels with respect to the different districts.

8.4.4.5 Cash journal (20 user)

The Munich Philharmony Orchestra uses an web application for cash journaling, they needed a slim, reliable accounting system instead of SAP licenses.

8.4.4.6 Artothek (15 user)

Similar to a public library the Artothek offers paintings and sculptures of young Munich artist for rent. The application manages the administration of pieces, rentings and users as well as the tracking and calendar management.

8.5 Conclusion

Ordo framework fits perfectly the needs of today's application development, especially in public administrations. Based on exclusively OpenSource tools it serves for reliable and free of license fee applications, all consistent in design, style and handling. The integration in existing infrastructure like databases or office suits guarantees the efficiency an optimal functionality. Especially with missing or poor user requirements a repetitive process including rapid prototyping the framework is a means to an end. The success in Munich shows the wide range of application running with the framework and should be repeated in other public administrations.

References

[EUPL - 1999]	http://joinup.ec.europa.eu/software/page/eupl
[Limux - 2012]	http://www.muenchen.de/limux
[Trinuts - 2012]	http://www.kitavm.de
[Phyton - 2012]	http://www.python.org/
[Apache - 2012]	http://httpd.apache.org/
[YAML - 2012]	http://www.yaml.de/

Appendix

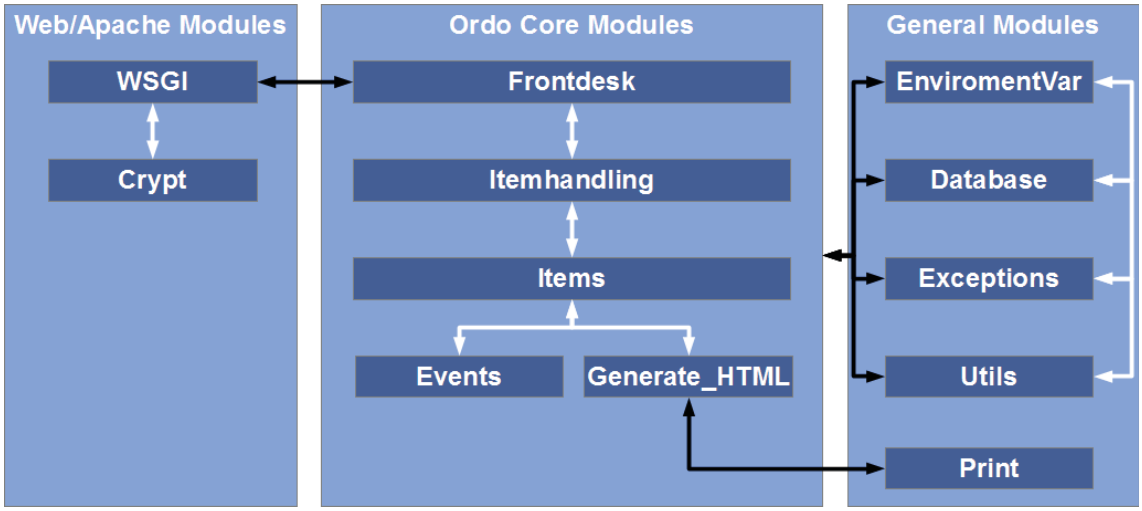


Fig. 1: Ordo Python modules structure

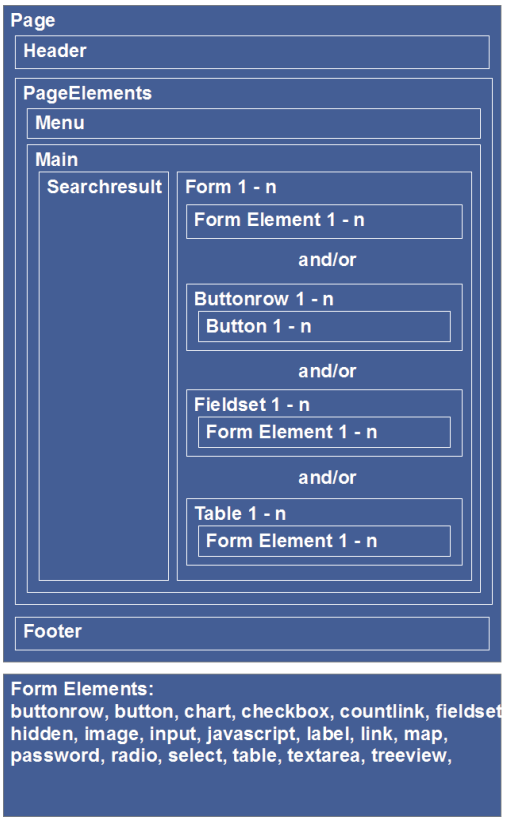


Fig. 2: HTML structure of Ordo pages

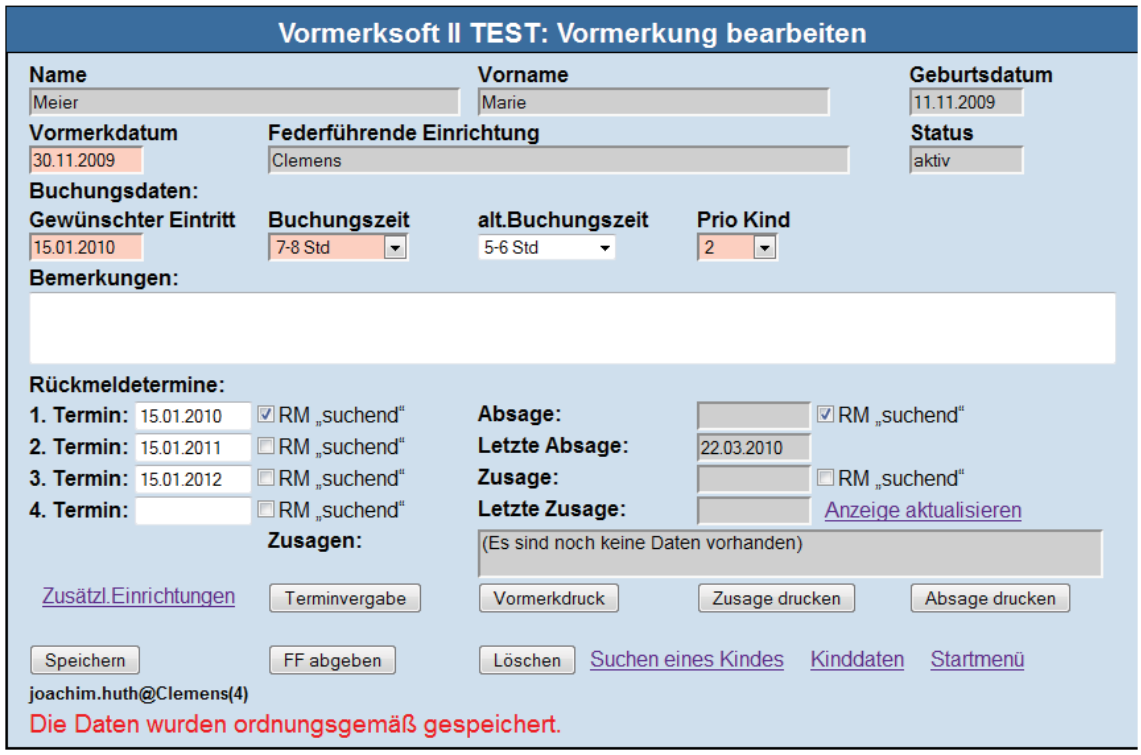


Fig. 3: Granting of play-school places, internal version in Munich

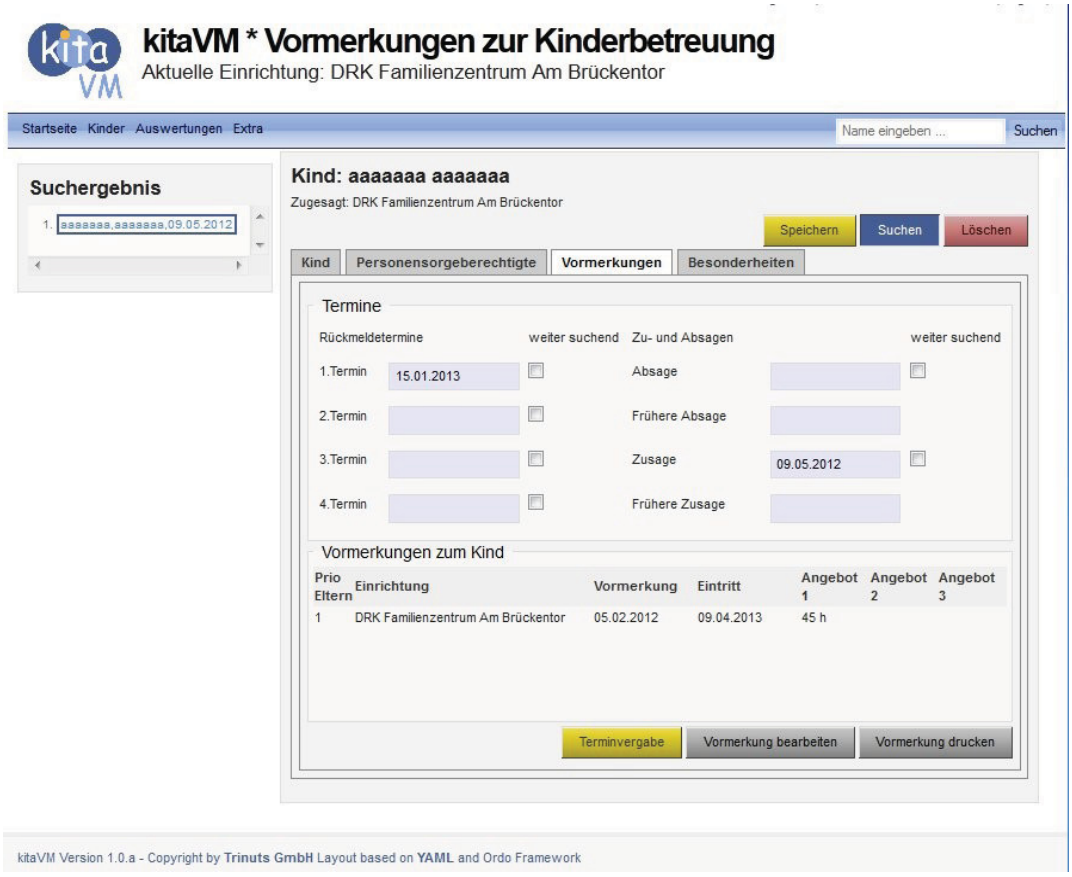


Fig. 4: Granting of play-school places, Ordo framework

9. The OpenJustitia Project at the Swiss Federal Supreme Court

Daniel Brunner

Federal Supreme Court of Switzerland, Lausanne/Switzerland

The IT of the Federal Court of Switzerland is aligned on the needs of the judges and the clerks. But the jurisdiction databases are also of interest to the attorneyship.

The Federal Court has the legal duty of informing the public about its jurisdiction. It performs this function by publishing its jurisdiction, among others, on the Internet website.

Moreover, the Federal Court is obliged to use its resources effectively and efficiently. In the IT-domain, it fulfills this aim by using consequently an open source strategy. According to this strategy, it has published the source code of its open source application OpenJustitia. This software, prepared for the courts, could also be useful for similar domains.

These terms of use and the modules of OpenJustitia will be described in the second part of this article.

9.1 Implementation of the open source strategy of the Federal Court

The open source strategy of the Federal Court contains two facets:

On one hand, the Federal Court uses open source applications which are normally at least as efficient as the proprietary applications: they are very secure, less vulnerable to viruses and resistant towards cyber attacks. They are also cheaper thanks to reduced or even non-existent license fees.

On the other hand, the Federal Court develops open source software itself to cover its own needs. It publishes this software under an open source license to lower taxes for citizens (not every court hast to reinvent the wheel) and to achieve synergy enhancements. Latter are realized by using extending the application with additional modules, which users of this application can develop to cover their own needs.

9.2 Usage of open source applications

The Federal Court uses, among others, the applications OpenOffice for the text processing, "Evolution" and "Thunderbird" for e-mails and shared calenders and especially "Alfresco", an extensive document management application.

9.3 OpenJustitia: an internal development, based on an open source software

The Federal Court, being the supreme court of the country and providing a uniform application of the law and a legal certainty in a trilingual area, has specific needs concerning jurisdiction databases. The development of an own jurisdiction database was indispensable, as there existed no product on the market, which satisfied the exigencies of the Federal Court. Therefore, it was necessary to develop the open source software OpenJustitia, which is based on other open source softwares like "Alfresco" and "Lucerne".

9.4 The OpenJustitia community

The source code of OpenJustitia has been published under the open source license "GNU General Public License Version 3" (GPLv3), on the end of September 2011. This procedure is in accordance/ conformity with the e-government strategy of the Confederation, the cantons and with the open source strategy of

the Federal Administration. The strategy was approved by the Federal authorities and has been well accepted by potential users. The OpenJustitia community has already thirteen members, including six IT-companies, representatives of 14 cantons and also a German institution leading in legal informatics. More adhesions are announced.

No competition and equal treatment are guaranteed.

The IT of the Federal Court is not about to become a new supplier of IT-services. It will not be in competition with private market participants. All services, which are associated to the integration of OpenJustitia into the IT-environment of other users, will be supplied by private companies, as soon as the know-how concerning the OpenJustitia software will have been transferred to the first five interested members of the OpenJustitia community. This point and the principle of equal treatment to all who are interested in it, are recorded in the rules of the OpenJustitia community.

9.5 The modules of OpenJustitia

At the moment, OpenJustitia contains different modules to manage court decisions and further documents and to complete and enrich them with additional meta data:

- OpenJustitia Doc: the main module to search court decisions and their management in various data collections;
- OpenJustitia Ldoc: searching in local and internal documents (requires OpenJustitia Doc);
- OpenJustitia Norm: automatic or semi automatic recognition of Federal article of laws in any documents;
- OpenJustitia Anom: anonymisation of court decisions with a possible linkage to a business management software;
- OpenJustitia Bib: searching in literature (requires OpenJustitia Doc);
- OpenJustitia Spider: integration of external legal data sources.

9.6 Future aspects

OpenJustitia, especially the module OpenJustitia Doc (searching of court decisions and other documents) has been conceptualised for the needs of the Federal Court. Other courts and institutions, which would like to compose their own data collection, could take into consideration to use the open source software OpenJustitia to do researches. For this purpose, they have to mandate an IT-company to integrate this software into their own It-environment and to supplement it with new specific functionalities (according to their needs).

Speakers

Omar Mansoor Ansari

Omar Mansoor Ansari
President of the
National ICT Alliance of Afghanistan
oansari@nictaa.af

www.nictaa.af - www.roscca.org

Open Source Afghanistan (OSA)
President of TechNation. LLC

- entrepreneur in Afghanistan‘s Information Communication Technology (ICT) sector
- founded Open Source Afghanistan (OSA), co-founded Open Source Alliance of Central Asia (OSACA) and Regional Open Source Conference of Central Asia (ROSSCA) to promote and provide a common platform for the Free/Open Source Software users in Afghanistan and in the Central Asia
- currently runs TechNation, an Information technology firm, as its full time president

Frank B. Brokken

Dr. Frank B. Brokken
Security Manager / Lecturer
f.b.brokken@rug.nl

Center of Information Technology
University of Groningen, Groningen/Netherlands

www.rug.nl

- Lecturer at Universities with considerable experience in teaching computer languages (C, C++, Java, Assembler, Pascal, Algol)
- Extensive background and active involvement in the realization of several large scale C++ software projects
- Broad experience in promoting and using Open Source software, in particular Linux

Daniel Brunner

Daniel Brunner
CTO, Head of Software development, help-desk, marketing, training and architecture
daniel.brunner@bger.admin.ch

Federal Supreme Court of Switzerland - Tribunal fédéral suisse
Av. du Tribunal fédéral 29,
1000 Lausanne 14 - www.bger.ch

- IT Head assistant, CTO, Head of Software development, Head of support and training team at the Federal Supreme Court of Switzerland
- Project Manager of OpenJustitia (www.openjustitia.ch)
- Specialist of Open Source Software

Wolfgang F. Finke

Prof. Dr. rer. pol. habil. Wolfgang F. Finke
Business Management - Business Information Systems
wolfgang.finke@bw.fh-jena.de

Ernst-Abbe University of Applied Sciences
Faculty of Business Administration
Business Information Systems Group
Carl-Zeiss Promenade 2
07745 Jena/Germany
www.winf.bw.fh-jena.de
www.wffinke.de

- IT manager/management consultant at Dornier/Daimler-Benz Group
- manager and client services director at Lotus Development Corporation (IBM) for Switzerland and Germany
- teaching and research at different European and North-American universities
- Ernst-Abbe University of Applied Sciences – professor of business management and business information systems
- Balkh University, Mazar-e-Sharif/ Afghanistan – two years assignment – cooperation with several public Afghan universities
- research focus: e-collaboration, e-learning, business process design, educational management, open source systems

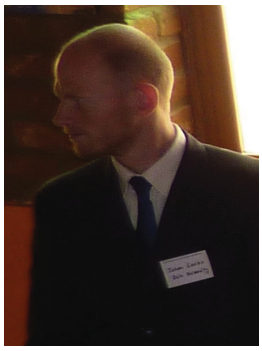
Sayed Jahed Hussini

MSc. Sayed Jahed Hussini
Lecturer, IT Manager
hussini@jahed.me

Balkh University
Faculty of Computer Science
Martyr Massoud Blvd.
Mazar-e-Sharif/Afghanistan
www.ba.edu.af/en

- Lecturer and IT Manager at Balkh University
- MSc. in Computer Science at the Technical University of Berlin, Germany

Johan Ivar Sæbø



Johan Ivar Sæbø
Researcher and implementer, Health Information Systems Programme (HISP)
johansa@ifi.uio.no

University of Oslo, Oslo/Norway
Faculty of Mathematics and Natural Sciences
Department of Informatics
Gaustadalléen 23B, Ole-Johan Dahls hus
0373 Oslo
www.mn.uio.no/ifi/english/

- Health Metrics Network/ WHO: Junior Professional Officer (JPO), Geneva/ Switzerland
- Botswana Ministry of Health/ BEANISH project, Botswana: Health Information System Consultant
- Cuban Ministry of Health/ University of Oslo, Cuba: Health Information Systems Researcher

Sundeep Sahay

Prof. Dr. Sundeep Sahay
Health Information Systems
sundeep.sahay@yahoo.com

University of Oslo, Oslo/Norway
Faculty of Mathematics and Natural Sciences
Department of Informatics
Gaustadalléen 23B, Ole-Johan Dahls hus
0373 Oslo
www.mn.uio.no/ifi/english

- Professor of Informatics at the Department of Informatics, University of Oslo, Norway, since 1999
- in 2012 co-authored the book titled "Integrated Health Information Architecture: Power to the users" (with his colleague Jorn Braa)

Frank Siebert

Dr. Frank Siebert
Member of the Board
Freies Office Deutschland e.V., Germany

www.frodev.org
fsiebert@trinuts.de

Member of the Board
Trinuts GmbH, Munich/Germany
www.trinuts.de/

- OpenSource software in public administration
- Project Management
- System Management (ITIL)

Matthias Stuermer

Dr. Sc. Matthias Stuermer
Member of the board of Swiss Open Systems User Group /ch/open
matthias.stuermer@ch-open.ch
www.ch-open.ch

Manager at Ernst & Young in Switzerland
www.ey.com/CH/en/home

Working group leader “Office Interoperability” of the Open Source Business Alliance OSBA
www.osb-alliance.de

- Manager specialized on social media governance, open source software and open government data
- member of the board of Opendata.ch
- project leader of www.opensource.ch

Michael Weinhara

Michael Weinhara
Hospital Economist, international team leader for EU and KfW projects, in health information and health management
michael.weinhara@epos.de

Senior Consultant Health Information Systems
EPOS Health Management GmbH
Hindenburgring 18, 61348 Bad Homburg/Germany
www.epos.de

Member of
HL7 - www.hl7.org
DGQ - www.dgq.de
AGEG Associate - www.ageg.de

- Team Leader/ Consultant – Construction of tertiary, secondary and primary care Health Facilities in North Afghanistan (Funded by KfW)
- Consultant – Development of the National Health Information Master plan for Vietnam 2011 – 2015 (Funded by EU)
- Consultant – Introduction of Health Information Systems in Primary Care Settings in Indonesia (Funded by EU)

Research Papers

Wirtschaftswissenschaftliche Schriften

Jahrgang 2013

Hussini, S. J., Finke, W. F., 2013, Workshop Proceedings, Sept. 15-17, 2012 Kabul/Afghanistan – Creating Awareness fort he Use of OpenSource Systems in the Public Sector in Afghanistan, Wirtschaftswissenschaftliche Schriften Heft 2/2013, Fachbereich Betriebswirtschaft, Ernst-Abbe-Fachhochschule Jena

Fröhlich, J., 2013, Erarbeitung eines Konzeptes zur Einführung bzw. Umsetzung eines Betrieblichen Gesundheitsmanagements in die unternehmerische Praxis auf der Grundlage von Erfolgsfaktoren des „Thüringer Netzwerkes Betriebliches Gesundheitsmanagement“, Wirtschaftswissenschaftliche Schriften Heft 1/2013, Fachbereich Betriebswirtschaft, Ernst-Abbe-Fachhochschule Jena

Jahrgang 2012

Albrecht, S., Bark, B., Beyer, C., Blossey, B., Bösnecker, P., Brandt, T., Buerke, G., Chen, L., Domnik, M., Gätcke, J., Schleicher, St., Schultheiß, S., 2012, Auszubildende im Handwerk – Eine empirische Studie in der Region Jena, Wirtschaftswissenschaftliche Schriften Heft 4/2012, Fachbereich Betriebswirtschaft, Ernst-Abbe-Fachhochschule Jena

Ernst, D., Stoetzer, M.-W., 2012, Beschäftigungseffekte von Innovationen auf Unternehmensebene: Ein Überblick theoretischer und empirischer Befunde, Wirtschaftswissenschaftliche Schriften Heft 3/2012, Fachbereich Betriebswirtschaft, Ernst-Abbe-Fachhochschule Jena

Rauschenbach, M., 2012, Strategisches IP-Management im Unternehmen unter Berücksichtigung rechtlicher Rahmenbedingungen, Wirtschaftswissenschaftliche Schriften Heft 2/2012, Fachbereich Betriebswirtschaft, Ernst-Abbe-Fachhochschule Jena

Buerke, G., 2012, Ausgewählte Aspekte zu Corporate Social Responsibility bei deutschen Großunternehmen und Nachhaltigkeit bei produzierenden KMU in Sachsen und Thüringen, Wirtschaftswissenschaftliche Schriften Heft 1/2012, Fachbereich Betriebswirtschaft, Ernst-Abbe-Fachhochschule Jena

Jahrgang 2011

Geyer H., 2011, Besonderheiten bei der Finanzierung von Wohnungsgesellschaften – unter besonderer Beachtung der Risiken aus Zinsänderungen, Wirtschaftswissenschaftliche Schriften Heft 2/2011, Fachbereich Betriebswirtschaft, Fachhochschule Jena

Lehmann, M.-Ch., 2011, Demografischer Wandel – Identifikation von personalwirtschaftlichen Handlungsfeldern auf Basis einer Altersstrukturanalyse am Praxisbeispiel Commerz Real AG, Wirtschaftswissenschaftliche Schriften Heft 1/2011, Fachbereich Betriebswirtschaft, Fachhochschule Jena

Jahrgang 2010

Döring, S., 2010, Zusammen flexibel ist man weniger allein? Eine empirische Analyse der neuen Arbeitsform Coworking als Möglichkeit der Wissensgenerierung, Wirtschaftswissenschaftliche Schriften Heft 1/2010, Fachbereich Betriebswirtschaft, Fachhochschule Jena

Jahrgang 2009

Pfeil, S., 2009, Kritische Darstellung der theoretischen Grundlagen zum Bildungscontrolling bei verhaltensorientierten Personalentwicklungsmaßnahmen, Wirtschaftswissenschaftliche Schriften Heft 2/2009, Fachbereich Betriebswirtschaft, Fachhochschule Jena

Krähmer, Ch., Stoetzer, W.-M., 2009, Die Nachfrageeffekte der Hochschulen in Jena - Eine Regionalökonomische Analyse der Einkommens- und Beschäftigungswirkungen, Wirtschaftswissenschaftliche Schriften Heft 1/2009, Fachbereich Betriebswirtschaft, Fachhochschule Jena

Jahrgang 2008

Dietmann, C., 2008, Kann denn Siegen Sünde sein? Die Ökonomik des Dopings am Beispiel des Radsports, Wirtschaftswissenschaftliche Schriften Heft 1/2008, Fachbereich Betriebswirtschaft, Fachhochschule Jena