Documenting Open Source Migration Processes for Re-use

Aurona Gerber
Meraka Institute of the CSIR
Brummeria
Pretoria, South Africa
agerber@csir.co.za

Onkgopotse Molefe
Meraka Institute of the CSIR
Brummeria
Pretoria, South Africa
omolefe@csir.co.za

Alta van der Merwe
Meraka Institute of the CSIR
Brummeria
Pretoria, South Africa
avdmerwe3@csir.co.za

ABSTRACT

There are several sources that indicate a remarkable increase in the adoption of open source software (OSS) into the technology infrastructure of organizations. In fact, the number of medium to large organizations without some OSS installations, is surprisingly low. This move to open source (OS), as well as the obvious advantages thereof, have motivated the CSIR of South Africa to investigate the adoption of OSS across the institution for all aspects of its operations. In launching this endeavor, it became apparent that there are very limited resources available, locally or internationally, that documented process related information about organizational OS migrations. This lack of information provides the motivation for this research that investigates the use of process reference models to capture the process related information for an organization-wide migration from proprietary software to OSS. In order to develop the necessary process reference models, the specific process models for the CSIR OS migration were captured, and, using a repeatable method based on reference model criteria, the generic process reference models for an organizational OS migration were extracted and documented. It is our firm belief that these process reference models would provide a baseline for the processes needed when any organization considers open source adoption or organization-wide OS migration.

Keywords

Process reference models, re-use of business process models, organizational open source migration.

1. INTRODUCTION

Several sources indicate a remarkable increase in the adoption of open source software (OSS) into the technology infrastructure of organizations [19]. In fact, the number of medium to large organizations without some OSS installations, is surprisingly low [26, 8]. Furthermore, acquisitions of several open source (OS) firms that recently made headlines indicate the growing significance of OSS in organizations today (Xen Source for $500 million; Zimbra for $350 million; Sleepycat for an undisclosed amount and JBoss for $350 million) [2, 22].

OSS is software that could be acquired at little or no financial cost, and it is licensed in a manner that allows users to study it, edit or improve it and redistribute it without having to pay any royalties to those that developed the software in the first place [1]. OSS is produced by a self-organized community that engages informally, usually online, in a way that represents the antitheses of a proprietary software strategy [14, 15]. OSS engages as many as possible collaborators and thus the approach is to maximize adoption throughout the value chain rather than using formal intellectual property rights (IPR) protection to set boundaries between vendors, their competitors and customers [22, 41].

Reasons cited for OSS adoption include the fact that is is free to use, copy and share, it is often more secure than any other proprietary operating systems, it can be customized to suit a particular business function or users needs, and it also support older hardware platforms eliminating the need for costly state-of-the-art equipment [8, 15, 17]. Within this paper, *OS adoption* refers just to the *adoption of OSS* as the software of choice, whilst *OS migration* refers to the *replacement of existing proprietary software with similar OSS solutions*.

In addition to the adoption of OSS in industry, several governments also investigated using OSS or even adopted policy to this effect [19, 38, 21]. After realizing some the benefits that OSS brings, South Africa took the initiative of introducing OSS migration projects within its governmental departments. One of the main incentives was the cutting of costs in the computing environment. The South African cabinet adopted the National Open Source Policy and Strategy in 2007. This decision was supported by numerous governmental organizations, who amongst others included CSIR, SITA, DST, WRC and CSPI who embarked on planning the use of OSS on their desktops and servers [9, 7].

The CSIR in its role as science partner of the South African government, launched a project to migrate the required information technology of all its operations, including desktops, to OSS platforms. The CSIR migration project to OSS was launched in June 2006 by the CSIR President, Dr Sibusiso Sibisi. The project was called *Vula*, which is a verb from the Nguni language meaning "to open" [39]. The decision was supported by the CSIR executive committee, CSIR employees and external stakeholders that CSIR collaborates with, for instance SITA and DST [13, 40].

After Project Vula commenced, it became apparent that there are very limited documentation available, locally or internationally, about the processes required for organizational OS migrations. There is a substantial number of publications about OSS characteristics and advantages, as well as literature about OSS development, but basically nothing could be found that systematically and formally discussed adoption processes. The prominence of OS at present causes organizations to consider its adoption on a organizational level, but the lack of documentation hinders this as organizations cannot estimate the effort and cost associated with OSS migration, and neither can they plan the migration project. Migration processes have to be determined from scratch when planning the OSS migration project.

This lack of information provides the motivation for this research that investigates the use of process reference models to capture the process related information for the CSIR's organization-wide migration from proprietary software to OSS. In order to develop the necessary process reference models, the specific process models for the CSIR OS migration were captured, and, using a repeatable method based on reference model criteria, the generic process reference models for an organizational OS migration were extracted and documented.

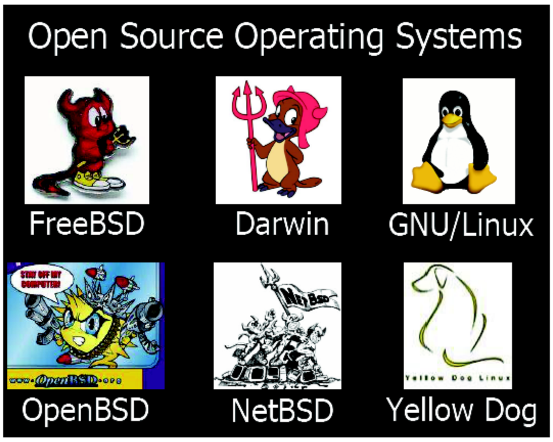
The paper is organized as follow: Section 2 provides necessary background about OSS, process modeling and process reference models. Section 3.1 discusses the capture of Project Vula's processes and Section 3.2 discusses the extraction of the process reference models from the documented Vula processes. Some reflections on the used approach are presented in Section 4, and the paper is thus concluded.

2. BACKGROUND

2.1 OSS and OS migration

As mentioned before, open source software (OSS) is defined as software that could be acquired at little or no financial cost, and it is licensed in a manner that allows users to study it, edit or improve it and redistribute it without having to pay any royalties to those that developed the software in first place [1]. In contrast to traditional proprietary software development practices, OSS is produced by a self-organized community that engages informally and that embraces a specific culture to promote freedom and the free use without commercial exploitation of software artefacts [14, 15, 41, 22]. OSS adoption has shown a remarkable increase recently and several governments also investigated OSS adoption [19, 38, 21].

Figure 1: Examples of open source operating systems



OSS are available both as operating systems (the operating environment of a computer) or supporting applications, which consist of programs such as word processors and web browsers used within the operating system [8]. Several OSS operating systems are available freely as is indicated in Figure 1. The most popular OS operating system used is Linux, which was developed by Linus Torvalds in 1991 who was a student at Helsinki University at the time [24]. Linux, a freely available operating system, is seen by its advocates as a replacement for the proprietary Microsoft Windows [38], proprietary operating system software where a user has to pay license fees for using it [8, 1]. Both Linux and Windows are available for desktops and server environments [24].

Dudley *et al.* [8, 23] used web searches to report an increase in the number of government departments, private (business) sector, academia (educational) and non-governmental organizations in South Africa that are migrating from proprietary software to OSS, mainly due to the high costs of license fees associated with proprietary software in South Africa. During OS *migration*, we are concerned with the replacement of proprietary software with OSS alternatives. Table 1 depicts common proprietary software programs together with their OSS alternatives [24]. The application software such as *Mozilla Firefox* and *OpenOffice* are available to run under both Windows and Linux and a user therefore does not have to migrate his operating system to make use of some OSS application software.

Worldwide, several governments also investigated using OSS in an attempt to curb costs [19, 38, 21]. The local as well as international government adoption also prompted the South African government as well as the CSIR, to investigate OSS migration [9, 7]. The CSIR, specifically, launched Project Vula with the aim to migrate its infor-

Table 1: **Examples of proprietary software vs. OSS alternatives**

Purpose	Proprietary Software	Open Source Software
Operating system	Microsoft Windows	Linux
Web browser	Internet Explorer	Mozilla Firefox
Office suite	Microsoft Office	OpenOffice.org
Image editor	Adobe Photoshop	GIMP

mation technology infrastructure, including desktops, to OSS in June 2006 [13, 40]. When Project Vula entered its planning phase, it became apparent that there were very limited documentation available, locally or internationally, about the processes required for organizational OS migrations. OSS adoption and migration is hindered by this lack of documentation because organizations do not have information available to estimate the effort and cost associated with the effort and cannot plan the migration project. These organizations have to plan the OSS migration project from scratch. This research therefore endeavors to document the migration processes of Project Vula in process models, as well as determine generic process reference models for reuse by other organizations when considering OSS migration. The next section discusses process modeling. Section 2.3 discusses process reference models and Sections 3.1 and 3.2 discussed Vula process models and the extraction of OS migration process reference models.

2.2 Process Modeling

Process modeling is viewed differently by various authors. Perumpalath [25] and Van der Merwe [37] define *process modeling* as *the procedure of constructing the process model using a standard notation.*Furthermore, BusinessRanks.com [5] refers to process models as *a graphical representation of the business processes, activities, actions, and operations that capture, manipulate, store, and distribute data between a system and its environment and among components within a system*, whilst Uvium [32] views process modeling as *the use of information and graphics to represent processes in a consistent way.* For the context of this study the definition of process modeling provided by Perumpalath and Van der Merwe will be adopted.

The primary purpose of process modeling is the documentation of process information [20]. Comprehensive documentation of processes as well as their activities could contribute to the success of many projects, especially if it is stored in a repository where it could be retrieved. Process models are used for process re-engineering, process re-organization, process monitoring and controlling, continuous improvement, quality management (such as ISO 9000), benchmarking, practice, and knowledge management [3, 37]. According to Childe *et al.* [6], process models are used to capture, track and analyze an organization's practices from the highest level down to the lower levels.

Both Childe *et al.* [6] and Van der Merwe *et al.* [36] mention that it is often difficult to identify exactly what constitutes a process, and that before striving to build process models, one needs to first identify exactly what a process, within a specific context, is. Building process model structures could be a complex and costly exercise [36].

The problems identified when constructing process models resulted in the notion of reutilizing *a set of generic process models* or *process reference models*. These generic process models form the baseline and could be adapted to be the process models that fit the specific environment of the organization [6]. Generic process models allow modelers to learn from the process designs of similar projects that were executed, which means that they do not have to build their own process models from scratch. This results in a saving of resources and effort [30].

2.3 Process reference models and their use

According to Rosa *et al.* [27], *process reference models* refer to *a set of reusable process model structures*, which could be used to capture the common activities, roles and resources of specific processes in a certain environment by adapting them to reflect the information necessary. Without process reference models the concept of process *design by reuse* could not be practiced effectively. Design by reuse promotes replication of existing processes that could enable companies to practice their business functions without having to design any of the available processes from scratch [34].

By definition a *process reference model* (also known as a *set of generic process models*, *a universal model* or *a set of process model patterns*) comprises of informative material in a library or knowledge repository regarding a set of generic processes discovered during a certain activity within a specific environment. These processes are generally represented as process model diagrams that graphically give an overview of the flow between the processes and their sub-processes [31, 37]. As a set of generic process models, process reference models are used to promote the reuse of existing process knowledge [35]. The main objective of a process reference model is to assist enterprises that has to perform similar processes with the reutilization of proven process knowledge [27]. Because of the emphasis on reuse, a process reference model reduces the risks and costs associated with repetitive errors of the same nature that tend to happen during the establishment of processes in a particular business or project [18]. Documenting process practices of the same nature in a given domain for reuse should provide consistent and satisfactory results to similar enterprises when these processes are adapted for their specific use [27, 18].

Van der Aalst *et al.* [33] compares *process reference models* with *plug-and-play* devices that can be used to plug-in to the process models of an enterprise, but they often require further improvement to reach perfection. Also addressing this issue is Tyrrell [30], who indicated that a working process needs to be monitored, improved and refined from time to time to ensure that it meets the requirements and possible purposes that it was initially intended for.

2.3.1 Documenting process reference models

At present process reference models are represented in various modeling languages and standard notations [10]. Prerequisites for a process reference model notation is that it is a widely accepted standard notation that can capture all the necessary information and that is preferably easily understood by modelers [27]. Notations like the **I**ntegration **DE**finition for **F**unction modelling (IDEF0) is widely used for the creation of process models [11]. IDEF0 is one of the serial set of standard reference methods for process modeling that the IDEF family of methods comprises of, and it was created by Wisnosky and Shunk. It is a standard notation that can be found and used both in Microsoft Windows and Linux for process modeling [28]. IDEF0 was selected for this research as the preferred notation for modeling the processes captured for the CSIR's migration to OSS.

Other commonly accepted process modeling notations used to present business processes or any other form of process information include **B**usiness **P**rocess **M**odelling **N**otation (BPMN), **E**nhanced **L**ine of **V**isibility **E**nterprise **M**odelling (LOVEM-E) and **A**rchitecture of **I**ntegrated **I**nformation **S**ystems (ARIS) [4]. Other modeling languages also sometimes used for specifying notations include **U**nified **M**odeling **L**anguage (UML) and the Ericsson-Penker business extensions [16].

Process reference models have been used by organizations to provide them with generic solutions to the process model construction. These solutions enabled different application domains (or organizations) to improve their business performance [34]. From the experiences of other organizations, it is evident that process reference models are one of the most powerful means to capture the acquired process knowledge of an organization [6]. This is why process reference models are the method used to disseminate the process knowledge the CSIR acquired during its OS migration in Project Vula.

3. OSS PROCESS REFERENCE MODEL EXTRACTION

In order to extract a set of process reference models for organizational OS Migrations it is necessary to execute the following tasks:

1. Identify the processes and compile all the process models of Project Vula.
2. Identify criteria for process reference models.
3. Extract a set of process reference models from the process models of Project Vula based on the established criteria.
4. Validate the extracted process reference models for organizational OS migration.

A systematic approach as suggested by van der Merwe and Kotzé [35] was used to execute the above mentioned tasks. This approach consists of five phases as is depicted in Figure 2 [35], namely: 1) Define scope, 2) Procedure selection, 3) Data gathering, 4) Comparison; and 5) Verification. The phases were executed as indicated below:

- Phase 1: Define Scope
  The objective of this study is to extract a set of process reference models for a typical OSS migration project of an organization. This could be used for the planning of future organization OS migration projects.
- Phase 2: Procedure Selection
  OSS migration projects were not performed as popularly as they are now, thus none of the procedures (or documented set of process models) exist for this type of migrations. Therefore a procedure utilized for the CSIR migration to OSS was to identify and capture process models using a standard process model notation (IDEF0).
- Phase 3: Data-gathering
  A document analysis, interviews and questionnaire were used as data collection tools for data gathering. The document analysis was done by reviewing all the project documentation, communication (or any other material) that had to do with the migration such as the data collection questionnaires and responses, which were captured and analysed by the Vula project team. Interviews were conducted with project managers, project team members and users who were involved in the migration. Interviews were held pre-migration as well as post-migration. Interviews with numerous migrated users provided information about their OSS experiences. All the questions and responses of interviews were recorded and transcribed, and are available as project documentation.
- Phase 4: Comparison Criteria for process reference models were compiled from literature.The set of Vula process models were compared against the criteria for process reference models and the set of generic process reference models were extracted.
- Phase 5: Verification
  This phase verified the extracted process reference models. For this study the models were verified using domain experts and project leaders.

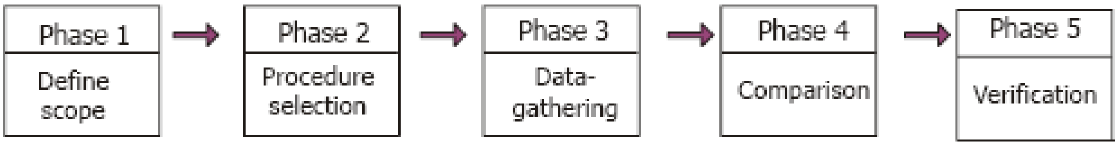
It should be noted that the scope of this study remained Project Vula, and the set of process reference models were only verified using domain experts. The models were not verified by implementing a new organizational OS migration, which would have been ideal, but it was beyond the scope of this project.

With regards to epistemological stance, this research was executed as a qualitative empirical study. The process documentation were primarily captured through observation and there was no interference into the project's execution. The migration was observed in the working environment of the project team and project execution office.

3.1 Vula Process Modelling

The CSIR's Project Vula was motivated by several reasons (i) to be one of the largest organizations to have adopted OSS in SA, (ii) to share the knowledge of the migration process acquired (or data collected) during the project with the public instead of keeping it confidential,

Figure 2: A systematic approach for process reference model extraction (Van der Merwe and Kotzé [35])



(iii) to enable other organizations to execute more effective OSS migration projects, (iv) to remove uncertainty with regards to aspects about the adoption and usage of OSS, (v) to empower users and scientists, (vi) to foster local ICT skills development and (vii) to further socio-economic development [21]. In addition, CSIR with more than 4000 employees, should save substantially on annual proprietary software licensing fees [12]. It was envisioned that Project Vula will not only benefit the CSIR, but it should also act as an opportunity to educate (or develop) emerging young software developers in OSS acquisition and development. OSS allows opportunities for software developers to experiment and contribute innovative new functionalities. In addition, OSS create opportunities for collaboration with other international open source communities and does not require a large investment [1].

In order to support the knowledge sharing of Project Vula, it is a prerequisite that the project processes enabling the OSS migration are documented. This is the first step necessary for the generation of generic process reference models. To construct a process model, it is necessary to first identify the top-level processes. After the top-level processes were identified, the necessary information about these processes are captured i.e. specifying various activities performed as part of the process, the order in which the activities have to be executed, inputs and outputs as well as responsible roles and goals [30, 29].

As stated, the data-collection methods were used for gaining knowledge about the activities of the migration to OSS, and this was used to identify the processes. The detailed steps in Table 2 were followed (or utilized) to identify and capture process models for the OSS migration project at CSIR.

As specified in Table 2, the high-level diagram is called a parent diagram. This diagram consists of main processes that are broken down into sub-processes in lower-level diagrams. This decomposition continues until atomic processes that cannot be further decomposed, are identified. In this case the parent diagram will be called a high-level process model diagram, while the decomposed diagrams will be referred to as lower-level process model diagrams. Subprocesses and atomic processes represent a set of refined diagrams (lower-level process models), which were extracted from the high-level process models.

The second step refines the high-level process model diagram further into lower-level process models or sub-processes, sub-subprocesses or atomic processes. These processes include defining, identifying, capturing and decomposing critical migration tracks which were followed as part of the migration plan during the project. It was during this process that the Vula project website and documentation were utilized as resources to collect data and to further confirm all migration processes that took place.

3.1.1 High-level Process Diagram

The first step in Table 2 identified the high-level processes, which are the key or main migration processes of Project Vula. From 2006-2008, the lifespan of the project, the aim was to capture the essence of migration processes that other organizations can learn from. During the timespan of the project, formal and informal interviews (pre-migration and post-migration) were conducted with involved Vula project team members and CSIR employees. The high-level process model diagram is presented in Figure 3, indicating the graphical representation of process names, inputs and outputs of summarized migration processes. Table 3 name each process and list inputs, outputs and goals.

The high-level process model diagram in Figure 3 indicates how an output of one process becomes an input of another process. For example, for the *Kick-start the project (A1)* process. the *CEO Declaration* is identified as an input resource and *Project initiated* is the output resource. *Project initiated* then becomes the input resource for the next process *Form the project team (A2)*. It is possible to have more than one output in a process, and it is also possible that refined processes do not have an input resource. As depicted in Table 3 each process has its own goal, the *goal* of the *Kick-start the project (A1)* process is to provide assurance to the CSIR employees and its external stakeholders that the project had begun.

Process *Form the project team (A2)* has as purpose to ensure that a reliable and committed team is in place to plan the migration once the project has been initiated. The *Announce the project publicly (A3)* process's goal is to ensure that the public is informed about the project and kept up-to-date all the time with regards to its progress. The *Develop migration plan, divide the project into tracks (A4)* process was refined further into parallel processes or the five migration tracks namely *Execute communication track (A41)*, *Execute technology track (A42)*, *Execute training track (A43)*, *Execute roll-out track (A44)* and *Execute maintenance track (A45)*. All these processes' outputs are inputs into *Migrate scheduled users to OSS (A5)*. After users have been migrated, support and maintenance continued to be offered to users on a daily basis (*Support and maintenance (A6)*) and the migration knowledge was documented (*Document lessons learnt (A7)*).

The high-level process model diagram above as depicted in Figure 3, represent the key processes of the OSS migration project. The high-level process where decomposed further into sub-processes for which process models were

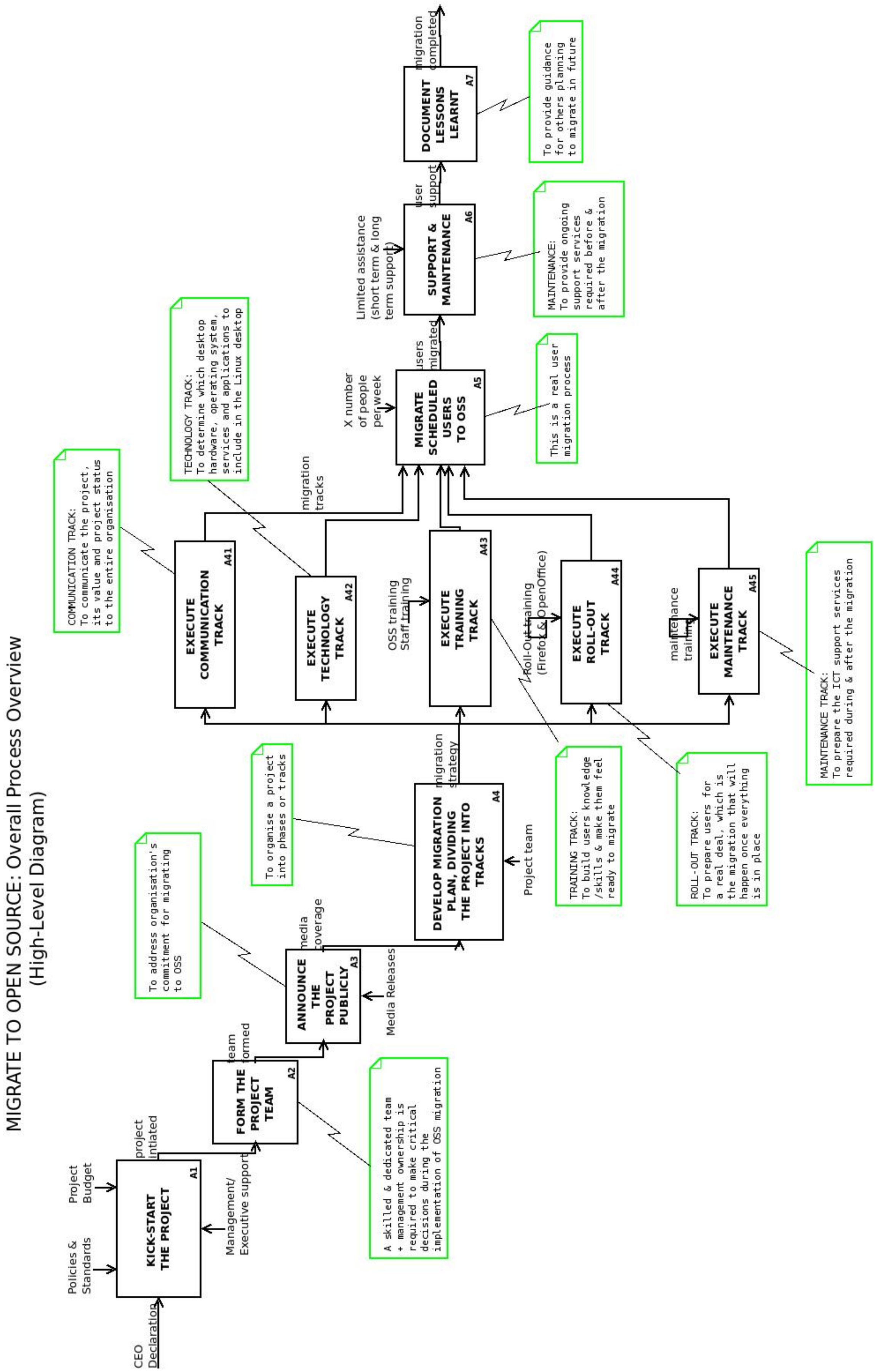
Table 2: The procedure used for capturing OSS migration process models

Step	Step description	Tools/documentation used	Deliverable
1	Derive the high-level process model	Process listing with goals and resources	High-level process model
2	Refine the high-level process model to subprocesses	Subprocess and atomic process (sometimes referred to as sub-subprocesses) listing	Subprocesses and atomic processes

Table 3: High-level processes input, output resources and goals

Process	Input/output resources	Goal description
Kick-start the project (A1)	Input: CEO Declaration Output: Project initiated	To prove the organization's seriousness and commitment towards migrating to open source
Form the project team (A2)	Input: Project initiated Output: Team formed	To make critical decisions during the implementation of an open source migration
Announce the project publicly (A3)	Input: Team formed Output: Media coverage	To ensure that the public knows about this type of a project
Develop migration plan, divide the project into tracks (A4)	Input: Media coverage Output: Migration strategy	To help draw a roadmap for the current environment to transitional environment thereby sub-dividing the project into tracks to allow much of the work to be done thoroughly by those given responsibility for the task
Execute Communication Track (A41)	Input: Migration strategy Output: Migration track	To create user awareness and excitement for changing to OSS
Execute Technology Track (A42)	Input: Migration strategy Output: Migration track	To check for alternative OSSs and the compatibility of such softwares against the current ones used by users
Execute Training Track (A43)	Input: Migration strategy Output: Migration track	To provide users with the relevant training and build their skills to make them feel confident about the migration
Execute Roll-out Track (A44)	Input: Migration strategy Output: Migration track	To prepare users and put them into action by installing some of OSS-related applications in their desktops
Execute Maintenance Track (A45)	Input: Media coverage Output: Migration strategy	To continue to provide all the help needed even after the completion of the migration
Migrate scheduled users to OSS (A5)	Input: Migration strategy Output: Users migrated	To deliver an operational Linux desktop
Support and maintenance (short term and long term) (A6)	Input: Users migrated Output: User support	To continue to provide all the help needed even after the completion of the migration
Document lessons learnt (A7)	Input: User support, process model Output: Migration completed	To provide guidance to other organization planning to migrate to another distribution in future on how to go about the migration and to avoid any risks involved

Figure 3: The OSS migration high-level process model diagram



compiled. This paper just present the high-level process diagram for illustrative purposes, the rest of the process models are available in the thesis at
http://uir.unisa.ac.za/dspace/handle/10500/3263.

3.2 Process Reference Models for Organizational OS Migrations

In order to extract process reference models from a set of process models, it is necessary to extract a set of criteria to which process reference models must conform. This was done through a document analysis study.

Process reference models should be *reusable* [27, 35]; generic [35]; and should provide enough information within a specific context about processes necessary to execute a specific project or business function [31, 37]. The scope should be well defined, as well as the outcomes and results. The process reference models should therefore provide a complete *set* of processes necessary to fulfill a specific objective [34].

The criteria for the extraction of process reference models are therefore:

- The context, goal and results should be clearly defined.
  For this study, the context is organization OS migration and the desired outcomes and results are an empowered and migrated organizational task force.

- The set of process reference models should be complete and thus enable the execution of a complete business function.
  Within project Vula, all the process models of a project being executed, were identified and documented. The generic processes were extracted from the process models, and therefore enables the execution of a complete business function.

- Process reference models are in nature *generic*, meaning *applicable to an entire class or group.* This criterion will be used to identify generic process models from the process models.

- Process reference models must be *reusable*, meaning that the process in its entirety can be reused within another context to achieve the same result. This criterion will be used to identify generic process models from the process models.

Using the high-level process diagram of Figure 3 as example, Table 4 shows how process reference models were extracted. The results for the high-level processes are depicted in Figure 4. This paper just present the high-level process reference model diagram for illustrative purposes, the rest of the extracted process reference models are available at
http://sites.google.com/site/ontologyprojects/home/generic-process-models-for-os-migration.

4. CONCLUSION

In this paper we discussed an approach for the extraction of process reference models for an organizational OS migration project. The CSIR embarked on a project (project Vula) to migrate the organization to OSS. In the planning phase of Project Vula, it became apparent that no process relation information for an organization OS migration exists in literature. As part of project Vula, it was decided to capture the necessary information in order to facilitate similar endeavors by other organizations.

In oder to capture the process relation information, it was decided to derive a set of process reference models for organization OS migration. Process reference models provide a set of baseline processes that could serve as a starting point for any organization that wants to perform a similar business function in future.

The process reference models were extracted from a set of process models that were constructed for Project Vula. This process models captures all the process related information of the project. From the set of process models, a set of process reference models were identified using set criteria for process reference models.

This study thus confirms that process reference models for an organization OS migration can be established. These models could be used to reduce uncertainty in organizations planning to migrate from proprietary software to OSS as it captures organizational learning in a way that could serve as a guide to help in planning and implementing an OSS migration project. It is our firm belief that these process reference models would provide a baseline for the processes needed when any organization considers open source adoption or organization-wide OS migration.

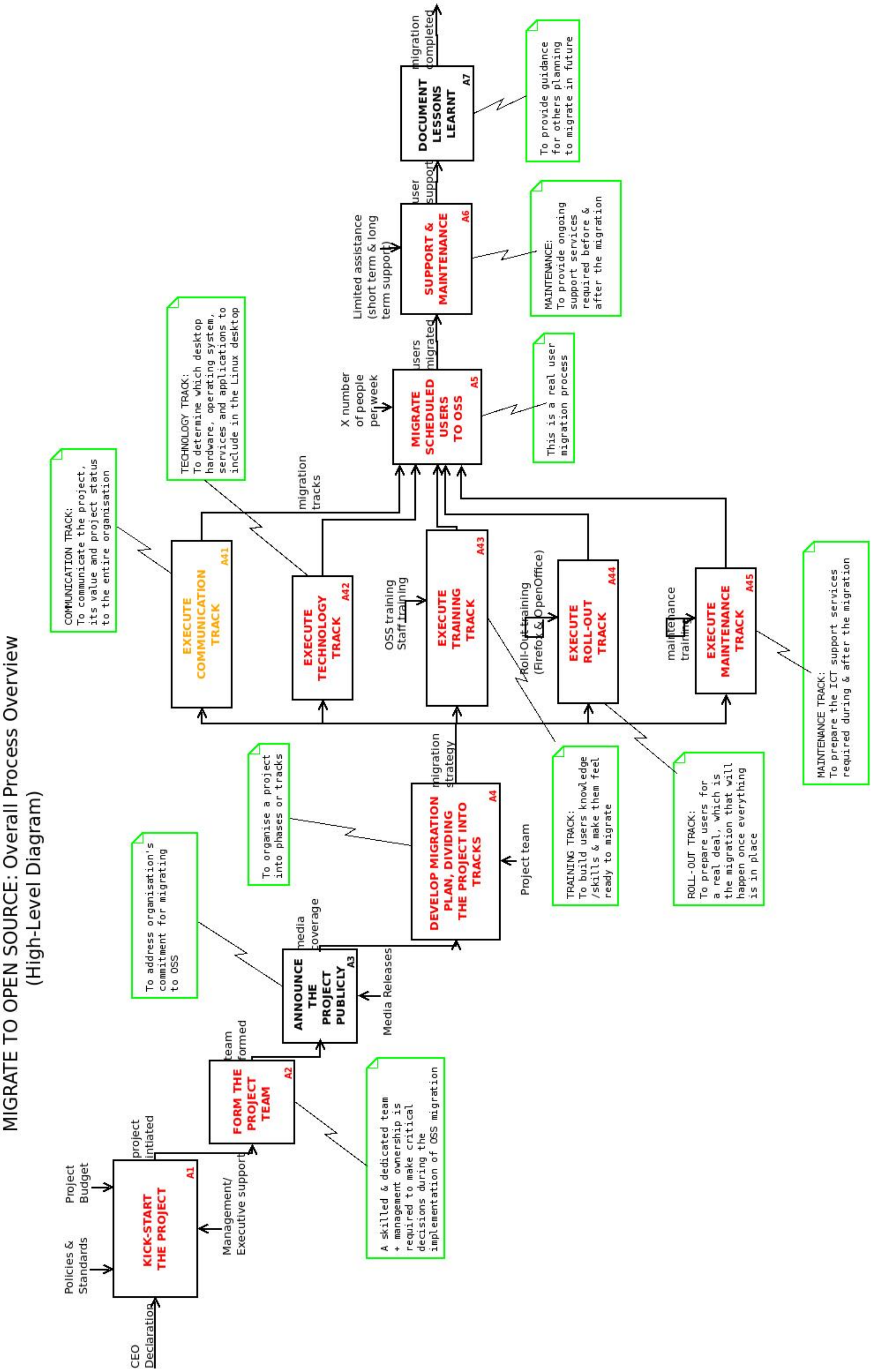
5. REFERENCES

[1] O. Ahmed. Migrating from proprietary to open source learning content management systems. 2005.
[2] C. Babcock. Open source for sale. InformationWeek, ABI/INFORM Global, pp. 25-30, 2008.
[3] B. Barn. Business process modeling. 2007.
[4] D. Brain, P. Seltsikas, and D. Tailor. Process modelling notations for egovernment: An assessment of modelling notations for identity management. 2005.
[5] BusinessRanks.com. Process modeling - process modeling resources, 2006. http://www.businessranks.com/process-modeling.htm. Last accessed 27/7/2007.
[6] S. J. Childe, P. A. Smart, and A. M. Weaver. The use of generic process models for process transformation. pages 1 – 10, 1997.
[7] dst.gov.za. Open source software freedom day celebrations at the department of science and technologyâĂŹs (dst), 2006. http://www.dst.gov.za/. Last accessed 02/10/2009.
[8] G. Dudley, A. Finlay, and A. Otter. Open source software: The five step migration guide, 2006. http://www.opensourceafrica.org/. Last accessed 25/02/2008.
[9] eGovEditorialTeam. Government backs open formats, 2007. http://www.epractice.eu/en/news/281947/. Last accessed 3/11/2009.
[10] P. Fettke, P. Loos, and J. Zwicker. Business process reference models: Proceedings of the workshop on business process reference models (bprm 2005). 2005.
[11] FIPS_PUBS. Announcing the standard for integration definition for function modeling (idef0). *Draft Federal Information Processing Standards Publication 183*, 1993.

Table 4: Extraction of process reference models from the high-level processes

Process	**Generic**	**Reusable**	**Reason**
Kick-start the project (A1)	Y	Y	The process is generic and reusable. Any OS project must be started.
Form the project team (A2)	Y	Y	The process is generic and reusable. To execute the OS migration, the project team must be formed.
Announce the project publicly (A3)	N	N	The process is neither generic or reusable This is not a necessary process for the migration to succeed.
Develop migration plan (A4)	Y	Y	The process is generic and reusable. To execute the OS migration, a plan for all tracks must be compiled.
Execute Communication Track (A41)	N	Y	The process is not generic but reusable. To execute the OS migration, communication is preferable, but not necessary.
Execute Technology Track (A42)	Y	Y	The process is generic and reusable. To execute the OS migration, this process is indispensable as it identifies the necessary OSS technology solutions.
Execute Training Track (A43)	Y	Y	The process is generic and reusable. To execute the OS migration, the project must ensure training for all participants.
Execute Roll-out Track (A44)	Y	Y	The process is generic and reusable. To execute the OS migration, the roll-out process is indispensable.
Execute Maintenance Track (A45)	Y	Y	The process is generic and reusable. To execute the OS migration, maintenance must be planned and executed.
Migrate scheduled users to OSS (A5)	Y	Y	The process is generic and reusable. To execute the OS migration, this process is indispensable as it executes the migration.
Support and maintenance (A6)	Y	Y	The process is generic and reusable. To ensure the OS migration is sustainable, this process must be executed.
Document lessons learnt (A7)	N	N	The process is neither generic or reusable. This process is not necessary for the OS migration to succeed.

Figure 4: The OS migration process reference models.
The process reference models are red. The possible process reference models are orange and the processes that do not form part of the process reference model set, remains black.



[12] T. Fogwill, H. Bezuidenhout, R. Verheij, and E. Basson. Notes on the current status of the csir migration to open source and open standards through project vula. pages 1 – 5, 2007.

[13] T. Fogwill, H. Bezuidenhout, R. Verheij, and E. Basson. Csir open source migration project status update. 2008.

[14] D. Foster. Maintaining a successful corporate community, 2008. http://fastwonderblog.com/2008/09/17/maintaining-a-successful-corporate-community/. Last accessed 23/10/2008.

[15] go opensource.org. Open source software, 2005. http://www.go-opensource.org/software/. Last accessed 17/04/2007.

[16] M. Havey. *Essential Business Process Modeling*. O'Reilly Media, Inc, 2005.

[17] J. Horstmann. Migration to open source databases. *Computation and Information Structures (CIS)*, pages 12–13, 2005.

[18] C. Jensen and W. Scacchi. Reference model based open source software process discovery. 2003.

[19] A. Martinez. Open source, a development option. 2005.

[20] F. Michael. Process modeling purpose, 2009. http://www.bpmnforum.net/blog/bpmn/bpmn-in-practice/process-modeling-purpose/. Last accessed 17/03/2009.

[21] T. Mokhema. Go open source task team conference results of 22 and 23 august 2005 - the go open source campaign partners, government departments, the private and public sectors and academia, 2005. http://www.meraka.org.za/news/. Last accessed 16/01/2009.

[22] N. Munga and T. Fogwill. An analysis of the value that open source contributes to business models. In *IST-Africa 2009 Conference Proceedings, Paul Cunningham and Miriam Cunningham (Eds). IIMC International Information Management Corporation*, number ISBN: 978-1-905824-11-3, 2009.

[23] Opensourceafrica.org. Open source software - the five step migration guide, 2007. http://www.opensourceafrica.org/.

[24] N. Peeling and J. Satchell. Analysis of the impact of open source software. pages 4 – 5, 2001.

[25] B. Perumpalath. Modelling business process: An integrated approach. 2005.

[26] RedHat_Inc. Open source adoption, 2007. http://www.redhat.com/. Last accessed 16/11/2007.

[27] M. L. Rosa, F. Gottschalk, M. Dumas, and W. M. van der Aalst. Linking domain models and process models for reference model configuration. 2005. Last accessed 25/02/2008.

[28] Y. Sun. An introduction to process modeling, 2006. http://yang-sun.com/modeling.aspx. Last accessed 13/02/2009.

[29] N. Turbit. Business process modelling overview, 2005. Last accessed 17/7/2007.

[30] S. Tyrrell. The many dimensions of the software process. 2000.

[31] US_Homeland_Security. Process reference model for assurance mapping to cmmi. pages 1 – 9, 2008.

[32] I. Uvium. Business process management and optimization glossary, 2005. Last accessed 03/02/2009.

[33] W. van der Aalst, A. Dreiling, F. Gottschalk, M. Rosemann, and M. Jansen-Vullers. Configurable process models as a basis for reference modeling - business process reference models position paper. pages 75–81, 2005.

[34] W. M. van der Aalst, A. Dreiling, M. Rosemann, and M. Jansen-Vullers. Configurable process models as a basis for reference modeling. 2003. Last accessed 26/01/2009.

[35] A. van der Merwe and P. Kotzé. A systematic approach for the identification of process reference models. 2008.

[36] A. van der Merwe, P. Kotzé, and A. Gerber. Generic process model structures: Towards a standard notation for abstract representations. pages 172–180, 2007.

[37] A. J. van der Merwe. Unisa thesis: Towards a reusable process model structure for higher education institutions. 2005. Last accessed 26/01/2009.

[38] V. van Reijswoud and C. Topi. Alternative routes in the digital world, 2003. http://opensource.mit.edu/. Last accessed 30/01/2009.

[39] VulaWiki. Project vula, 2006. https://vula.csir.co.za/. Last accessed 17/04/2007.

[40] VulaWiki. Vula progress report 1, 2006. https://vula.csir.co.za/. Last accessed 19/11/2007.

[41] J. West. How open is open enough?: Melding proprietary and open source platform strategies. *Research Policy: Open Source Software Development*, 32, Issue 7(ISSN 0048-7333):1259–1285, 2003.