

African Virtual Open Initiatives and Resources (AVOIR)

Internal Evaluation

Prepared for IDRC

7 October 2007

Contact:

Philipp Schmidt

phi.schmidt@gmail.com

Table of contents

African Virtual Open Initiatives and Resources (AVOIR)	1
Internal Evaluation	1
Table of contents	2
Introduction	3
AVOIR overview	4
IDRC Support for AVOIR.....	4
Project Timeline.....	5
Structure.....	5
Methodology	7
Main sources of data.....	7
AVOIR Framework.....	8
Consolidated Vision.....	9
Consolidated Mission.....	9
Findings	11
Overview Boundary Partners.....	11
Boundary partner: AVOIR nodes.....	11
Outcome Challenge	12
Progress Markers	13
Observations.....	14
Boundary partner: Software developers.....	17
Outcome Challenge	17
Progress Markers.....	17
Observations.....	19
Boundary partner: Users of the software developed by AVOIR.....	24
Outcome Challenge	24
Progress Markers.....	25
Observations.....	25
Additional boundary partners.....	26
Software development.....	27
Competitive environment.....	28
Conclusions and Recommendations	30
Challenges and recommendations.....	31
Recommendations.....	31
Annex A Outcome Mapping Overview	34
Annex B Developer mailing list analysis	36
Annex C Additional boundary partners	37

Introduction

This is an evaluation of the African Virtual Open Initiatives and Resources (AVOIR) network of African Universities. The approach this evaluation takes is to identify the strengths of AVOIR and how the project can build on them, and explore a few areas where we see room for improvement. In the process the evaluation revisits some of the original project goals, develops a clarified vision and mission, and creates a monitoring and evaluation methodology that could be used from this point forward. Hopefully this document will support the project in its effort to grow and be sustainable.

In a sense, this is – although forward looking – a retrospective evaluation. Typically a framework for monitoring and evaluation will be designed at the project start and is then used to monitor progress and highlight the need for changes in the strategy or implementation. Doing this backwards poses some problems. One problem is a lack of data, which we tried to make up for by using the existing progress reports that were prepared for IDRC, analysing mailing list statistics, and other data on the software development process. Secondly, over time the goals and objectives of a project become more distinct, and sometimes shift from the original stated ones. For this reason the original objectives from the Memorandum of Grant Conditions seemed too limiting, and we have tried to expand (but also clarify and combine) them.

It is useful for the reader to have some familiarity with the outcome mapping (OM) methodology and the AVOIR project. However, short definitions of OM terms are added whenever a new term is first used. Please see the Annex for an overview of Outcome Mapping and its terminology.

AVOIR Project Manager Abdul Fakier and AVOIR Researcher Enver Ravat participated in the design of the overall evaluation framework, including the project vision and mission, and the definition of boundary partners, and their outcome challenges and progress markers. Where appropriate the pronoun “we” is used to highlight this collaborative effort.

However, the evaluation of AVOIR against this framework, was done by Philipp Schmidt, and it does not necessarily reflect the opinions of the above, or any other AVOIR staff.

AVOIR overview

This section provides a very short overview of the AVOIR project, mostly for readers who are not familiar with it. For a more detailed description, please refer to the AVOIR website and the first Technical Report submitted to IDRC.

“The African Virtual Open Initiatives and Resources (AVOIR) project is a collaborative effort among several African higher education institutions to support capacity building in Free and Open Source software engineering. The main core activity of capacity building is undertaken through software design, development, deployment and support. AVOIR has four areas against which participation in the network is assessed: collaborating, developing, implementing and connecting. Collaboration involves participating in the shared activities of the network, which mostly happen online, supplemented by three developer workshops and one board workshop to date. Developing involves writing code according to shared coding practices that lead to the creation of modular software. Participants are responsible for implementing the software produced by the network, so most have implemented the KEWL.NextGen e-learning platform, and others are implementing other tools such as community forums, committee administration system, electronic thesis and dissertation system, alumni portal, and others. AVOIR encourages connecting with and establishing partnerships with local business, government and educational organizations, and the creation of local business opportunities.”

<http://avoir.uwc.ac.za/avoir/index.php?module=cms>, July 2007

IDRC Support for AVOIR

IDRC has supported AVOIR as a two year research study, running from 1 October 2004 to 30 September 2006 (with an extension until June 2007) in order to better understand how a multi national network like AVOIR can be established using “concepts of knowledge ecology” in the area of free and open source software development. The term “knowledge ecology” is not defined in the Memorandum of Grant Conditions, but can be understood as a system of different actors that, by interacting with each other (in many different ways, and taking on different roles) are creating a sustainable system.¹

“The overall objective of the research Project is to research how concepts of knowledge ecology can be used to build a sustainable and expanding system of free software creation that contributes to sustainable economic development in Africa.

This is a research and development project that aims to understand how to create software development in Africa as a collaborative process while at the same time creating that capacity in higher education institutions and producing software that can be used in the education sector as well as in the economic sectors.”

¹ This loose definition is based on conversations with Prof. Derek Keats and review of AVOIR project documents, which describe some of the different actors and their roles.

Memorandum of Grant Conditions between IDRC and UWC, Signed 20 October 2004

Project Timeline

The following table only contains some of the key milestones, see AVOIR website at <http://avoir.uwc.ac.za> for more detail on the individual items and a more complete list.

2004 October	IDRC Grant announced
2004 December	Release of alpha preview of KINKY application framework, and KEWL.NextGen e-learning system
2005 January	Developer workshop
2005 January	Release of beta version of KEWL.NextGen e-learning system
2005 May	KEWL.NextGen Release Candidate 2 released Development of additional applications, based on KINKY framework:
2005 October	Developer workshop
2005 December	A number of modules were developed for use in healthcare by the Jooste Dispensary, which demonstrates the variety of potential uses of the KINKY framework
2006 May	kGroups application for group based collaboration
2006 June	AVOIR participates in international open source developer survey
2006 June	AVOIR publishes principles and manifesto
2006 June	Training at Eduardo Mondlane University
2006 June	Online board meeting
2006 November	Developer workshop and board meeting
2007 February	On-site developer workshop at University of Dar es Salaam
2007 May	On-site developer workshop in Kabul, Afghanistan
2007 May	AVOIR workshop in Nairobi (for participants of the e-learning Africa conference)
2007 June	On-site developer training, and e-learning workshop at University of Jos
2007 July	On-site developer training at University of Ghana
2007 August	AVOIR Masters Program workshop takes place at UWC
2007 August	Kabul developers visit UWC for training

Structure

The following table and diagram represent the organisational structure of AVOIR.

AVOIR Board	Consists of members from each of the AVOIR nodes
AVOIR Executive	Executive Director Prof. Derek Keats (UWC), Marketing and Communications Officer Beda Mutagahywa (University of Dar es Salaam)

AVOIR Secretariat	Project Manager Abdul Fakier (UWC), Chief Software Architect Paul Scott, and AVOIR Researcher Enver Ravat (UWC)
AVOIR Nodes	12 Universities
AVOIR strategic partners	Organisations that contribute to the overall vision and AVOIR goals, including: SUN Microsystems, the Media, GeekCorps, etc.
Clients of AVOIR nodes	Clients that pay for the services and products that a specific AVOIR nodes offers, including: Consol Glass, SANORD, etc.
Funders	AVOIR has received funding from a number of different organisations both local and international, including IDRC, UNESCO, USAID, Carnegie Corporation of New York, the Department of Science and Technology in South etc.
Individual developers	Software developers collaborating on AVOIR projects, but not formally aligned with AVOIR nodes [See section on boundary partner software developers below]

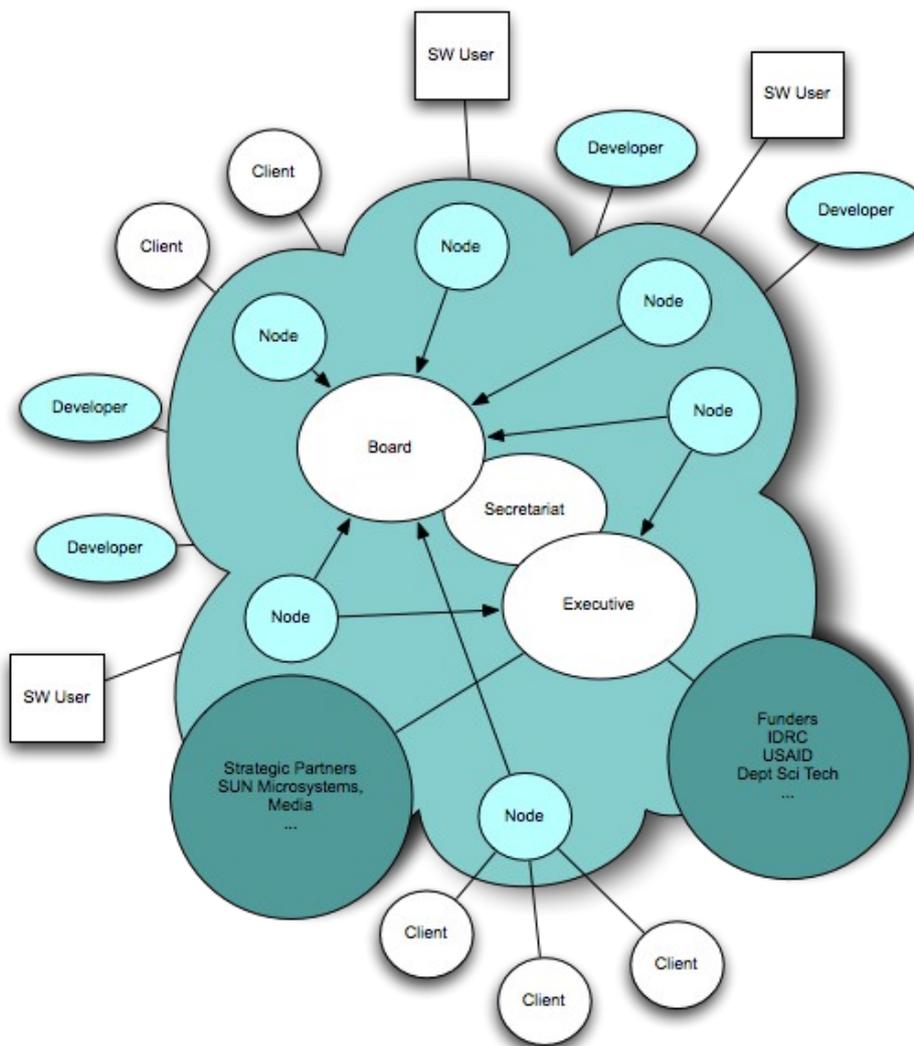


Illustration 1: AVOIR Structure

Methodology

This evaluation is based on a mix of outcome mapping (OM) and other evaluation strategies in this study. We worked through the OM planning steps retrospectively, used the original project proposal to define a vision and mission (which were missing); then identified boundary stakeholders, and described their outcome challenges and progress markers. However, we are not strictly following the OM terminology or orthodoxy – we might use terms that are not recommended in OM, for example *impact*, and we added a section on software development, which does not use OM.

Both the definition of outcome challenges and the description of project activities are aligned with an AVOIR internal framework developed during the November 2006 AVOIR workshop at UWC, which considers four areas of activity: *coding, implementing, collaborating, connecting*.

A draft of this evaluation was shared in two rounds of internal review to provide an opportunity to AVOIR to submit additional data that might be missing, and correct factual errors. The first draft was shared with Prof. Derek Keats, Abdul Hadi, Enver Ravat and Paul Scott at UWC. The second draft was shared with the AVOIR board members. Findings and conclusions of the evaluator take into account the comments received during these review rounds. If AVOIR team members disagreed with evaluation findings, these findings were not removed from the report, but differences in opinion highlighted.

Main sources of data

- Existing project documents, including the original grant agreement with IDRC, AVOIR principles and manifesto on the website, and various presentations by Prof. Derek Keats
- Interviews with developers during AVOIR workshop in Cape Town, November 2006
- Interviews with board members during AVOIR workshop in Cape Town, November 2006
- Conversations in person and by email with Prof. Derek Keats (AVOIR Executive Director), Abdul Fakier (AVOIR Project Manager), Paul Scott (Chief Software Architect) and other software developers at UWC
- Developer mailing list
- Board mailing list
- Data from the CVS system, which tracks software development activities
- Technical reports for IDRC
 - Report 1, September 2005

- Final report 2, September 2007, this report was prepared at the same time as the evaluation, which means that the evaluator has not had a chance to read all of it carefully and include it in this document.
- Conversations with e-learning team at UWC and other users of software produced by AVOIR; E-learning report to UWC senate.
- Documentation from the project website <http://avoir.uwc.ac.za>

AVOIR Framework

For its third workshop and to guide its activities, AVOIR developed a framework that describes activities in four areas: Coding, implementing, collaborating, connecting.²

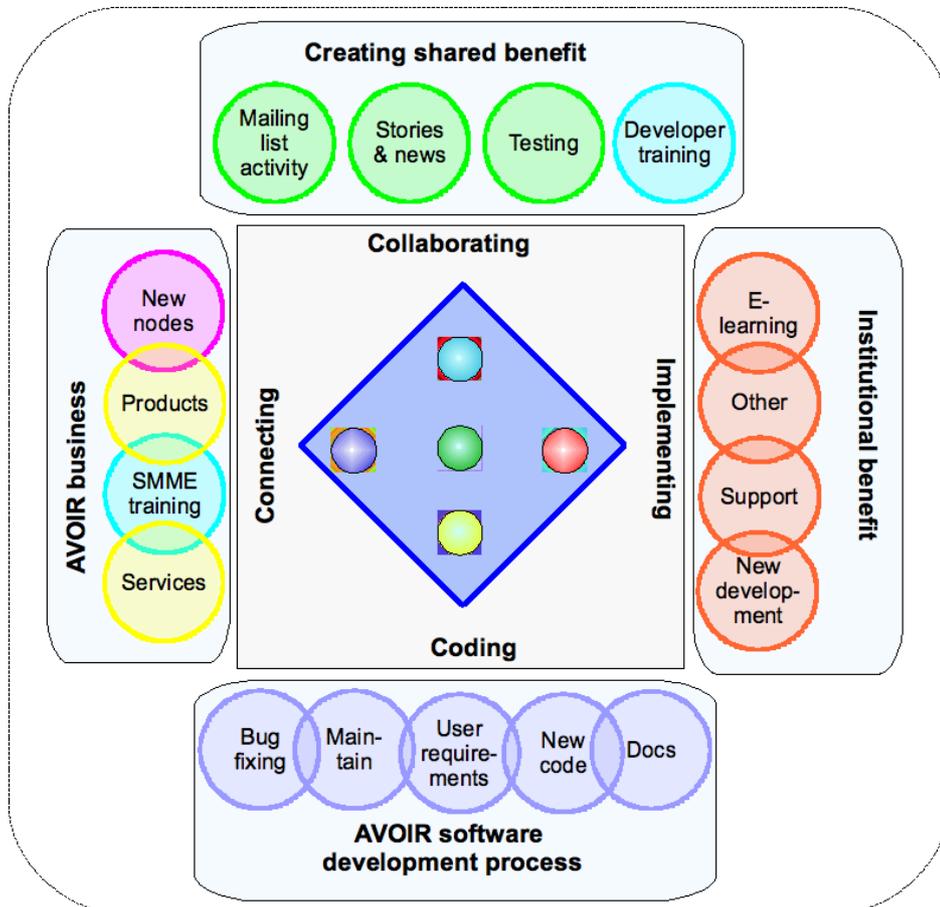


Illustration 2: AVOIR framework

During the second AVOIR workshop a comprehensive list of todos was created for all AVOIR nodes (organised by the four framework areas). The evaluator has not been able to access progress reports for these todos.

² This framework was only introduced during the November 2006 workshop, and according to the AVOIR Project Manager, not used consistently for monitoring and evaluation by the nodes.

Consolidated Vision

Outcome Mapping terminology: “The vision reflects the broad human, social & environmental betterment in which the program is engaged and to which it is contributing.”

The following vision statement was developed by the evaluation team. It aims to encompass all of the objectives listed in the Memorandum of Grant Conditions:

Globally, the concept of knowledge ecology is used to enable the sustainable development and diffusion of knowledge by networks of actors; organisations as well as individuals, from all sectors (public, private, civil society) collaborate to share and create knowledge.

Applying the concept of knowledge ecology, Africans develop and use world-class free and open source software (FOSS) in ways that are self-sustainable and contribute to economic development. Human capacity exists in Africa in the broad field of FOSS development, including marketing, software development, programming, project management. Knowledge is shared between an international community of developers in Africa and beyond. African organisations (both academic and private sector) successfully provide customisation, training, support and implementation of FOSS in Africa.

Consolidated Mission

Outcome Mapping terminology: The mission is one component within the vision, on which the program is going to focus, and where it can affect some change.

For this section, we picked the *key objectives* from the Memorandum of Grant Conditions and added the most important *specific objectives*:

In support of this vision, the project will:

- create a network of software development organised around, but not limited to, higher education in Africa;
- focus initial collaboration on development of an e-learning system (which takes into account low-bandwidth requirements in Africa) to establish the network and develop human capacity and promote the ideas of FOSS;
- build research capacity in researching the development, application and support of FOSS in Africa within African institutions;
- test and demonstrate how institutions (starting with higher education, but potentially other sectors) need to change in order to enable sustainable FOSS development;
- raise awareness among potential partner organisations to expand the network;

- use the network to understand (develop and test hypotheses) how ideas from knowledge ecology can create a sustainable system for Free and Open Source Software development, application and support; and present research findings as academic papers, contributions to books or at conferences.

Findings

Overview Boundary Partners

Outcome Mapping terminology: Boundary partners are those individuals, groups, and organizations with whom the program interacts directly to effect change and with whom the program can anticipate some opportunities for influence.

If AVOIR is successful in creating a FOSS ecosystem, then certain changes in behaviour and activities of its boundary partners will take place. This section will look at the most important groups of partners, list the kinds of changes that would indicate progress and describe to what extent we were able to observe these changes.

Applying the OM definition of boundary partners was difficult, because we were not sure what perspective to choose as our starting point. If we look from the point of view of UWC, we might be missing important issues in other nodes. If we consider boundary partners from the perspective of the network as a whole, we are venturing too far away from the real and practical issues that project nodes are dealing with individually. We ultimately decided to highlight this as a concern, but use UWC as the starting point of our exploration.

The key boundary partners of the AVOIR network are:

- AVOIR nodes
- Software developers
- Users of software developed by AVOIR

Additional boundary partners can be identified, but due to lack of data are not covered in detail.³

Boundary partner: AVOIR nodes

AVOIR is a network that consists of 12 nodes. Eight universities created AVOIR in November 2004, and three additional members joined in October 2005. The levels of activity, and feeling of ownership, at the AVOIR nodes determine the success of the network as a whole.

³ These include: Clients of services and products offered by AVOIR nodes; University management and administration. Also see Annex for more detail on how to include these boundary partners in subsequent evaluations.



Illustration 3: Map of AVOIR nodes (red) and collaborating partners (blue) provided by Prof. Derek Keats

According to AVOIR Executive Director Prof. Keats, collaborating partners have contributed in some ways, but are not formally part of the project (or funded by it)

Outcome Challenge

The programme intends to see all AVOIR nodes taking ownership of the project beyond initial funding. The nodes implement local policies, processes, and activities in order to institutionalise participation in AVOIR (AVOIR is not an add-on or an independent project, but becomes en-grained with the nodes activities). Nodes create communities of local developers who actively contribute to the design and development of the FOSS. Nodes implement AVOIR supported software within their institutions (for e-learning, student administration, etc.). Nodes market FOSS solutions and are building sustainable FOSS business opportunities around adaptation, implementation, and support, targeting public and private sector and non-profit organisations. They adapt and extend FOSS to address local needs and preferences, including localisation into local languages. Nodes integrate the ideas and practices of FOSS into the curricula and capacity-building at the university. Nodes raise awareness (using all types of media) of FOSS in general (and its contribution to development) and the AVOIR project and the local services and products that the node can offer in particular. Nodes communicate project status and progress, highlight problems they face and share experience and expertise

to solve them. AVOIR nodes post relevant news to the AVOIR web site. They share information on funding opportunities with other AVOIR nodes, and co-operate / collaborate as appropriate for tender and grant applications for FOSS development and projects. Nodes attract new AVOIR members in order to expand the network.

Progress Markers

The program expects to see that ...	Comments
... AVOIR nodes are establishing or supporting e-learning activities	Not all nodes are implementing AVOIR software for e-learning.
... AVOIR nodes are establishing development teams at their institutions	Developers have been hired at most institutions, but show varying degrees of participation
... board members at AVOIR nodes are utilising the board members mailing effectively to discuss new ideas, issues and solutions	List shows relatively low traffic and is largely driven by UWC. See analysis in governance section below.
The program would like to see that ...	Comments
... AVOIR nodes are approaching potential strategic partners, not only for the University, but for the network as a whole	UWC is very active and successful in building partnerships, but less initiative at other nodes.
... AVOIR nodes are actively supporting the capacity-building processes of their developers	Mixed levels of success between institutions. See capacity building analysis below
... co-ordinators at AVOIR nodes are actively encouraging the local developers to participate and collaborate on the mailing list	Most nodes do not have the same strong support structures that exist at UWC. Developers are working alone or in very small team and even senior developers are still building capacity themselves.
... AVOIR nodes are bringing new members into the network to become AVOIR nodes	The evaluator was not able to get any data on nodes who are actively soliciting new members.
... the AVOIR Board members distribute responsibilities between themselves	The majority of responsibility rests with UWC. See section on governance below.
The program would love to see that ...	Comments
... AVOIR nodes are developing and exploiting business opportunities, as the University itself, or by setting up spin-off companies	UWC has shown notable successes, but others nodes are less active. See business development analysis below.
... AVOIR nodes are localising (adapting to the local environment, including language of interfaces) the framework and application and implementing localised versions	Localisation of the user-interface of KEWL into Portuguese was undertaken in Mozambique.
... AVOIR nodes are expanding the ideas of FOSS to content and teaching, and start to leverage the network to develop and exchange course materials and other resources	So far, this has not taken place, but the AVOIR Masters programme will include a content component.

... the network has developed a strong distributed structure that does not depend on one institutions driving and animating the network

UWC is currently the crucial core to the network. See governance structure below.

Observations

Capacity building

Capacity building is one of the key objectives of the AVOIR network. Capacity building takes place both at the local nodes and within the network (across nodes). Most of the capacity development affects young software developers, who are joining the AVOIR project either during their studies or just after graduation. Capacity building has successfully combined different activities and strategies, including:

- Developer workshops that bring together developers from all nodes
- Mentored internships of junior developers at some nodes
- On-site focused training workshops taking place at individual nodes
- Ongoing knowledge sharing as part of the distributed software development process
- In addition, a distributed Masters Programme in Computer Science with Specialisation in FOSS is currently under development, and going through an accreditation process at the partner institutions. It has been accepted in principle by Uganda Martyrs University.

These are described in much more detail in the software developer section below.

Business development / fundraising by local nodes

A cornerstone of AVOIR's strategy to be self-sustainable is to connect with local business environments and provide products and services not only within the university setting, but also to firms, international organisations, etc.

The UWC node of AVOIR has been successful at attracting a number of clients for development and support contracts. Project Manager Abdul Fakier estimates the total budget for these contracts as US\$390,000.⁴ These include: Provincial Government of the Western Cape, South Africa; UNESCO Odimics and Greenstone; UNESCO Ocean Teacher; USAID (Afghan); Consol Glass; Labour Research Service (LRS); Southern African Nordic Centre (SANORD); UNESCO; USAID; San Jose State University (SJSU); Community Radio Siberia and Sierra Leone; Shuttleworth Foundation; etc.

Other AVOIR nodes have not been able to market business services based on AVOIR software as successfully.

One can speculate on some of the reasons why despite these obstacles, UWC has been able to generate income from services and development and others have not. UWC has been an active promoter of FOSS solutions and

⁴ Comments provided by Abdul Fakier to an earlier draft to this document.

strategies for many years and has developed a strong reputation in this area. The ideals of open sharing of knowledge, and collaborative innovation find support at the highest level of UWC management and are considered in official University policies. Further, UWC has been under more pressure to generate income, since it funds the majority of software development (see below), and this activity is not covered by the IDRC grant. Finally, the role of Prof. Derek Keats, Executive Director of the Information and Communication Services Department at UWC stands out. Prof. Keats has invested significant time and energy to building linkages to donors and the private sector in order to increase awareness and generate income.

There is no conclusive evidence why similar successes have not been achieved at other nodes, but the general difficulty of marketing information systems, and especially FOSS solutions in Africa presents an obstacle, that was reported by at least one node.⁵ Targeting the outsourcing market in developed countries was mentioned as one possible strategy, but given the lack of capacity in most nodes, such a strategy is unlikely to succeed in the short term. There is a lack of skills with respect to implementing FOSS business models, and less pressure to obtain additional funds, could also play a role. In addition, AVOIR produced software has strong competition not only from proprietary alternatives, but also from within the international FOSS community (see more on this in the section on software development below) and development aid organisations who are one of the primary target markets might have clear ideas what applications they want to use.

In addition, the low development activity at most nodes makes it difficult for UWC to start out-sourcing projects among the AVOIR members. Prof. Derek Keats reported that there is enough business interest to sustain AVOIR without donor funding, but that the nodes needed to demonstrate their ability to deliver working software code and services, before shared projects could be undertaken.

“If we had confidence in the nodes, then we could generate more than enough business to keep AVOIR going without aid funding.”

Prof. Derek Keats, Email to AVOIR Board, 4 October 2007

However, at least one node reported some business successes, albeit not based around AVOIR software. The Jomo Kenyatta University of Agriculture and Technology (JKUAT) in Nairobi established a Software Development Unit with the support of AVOIR, which has participated in a number of projects for both internal clients (website and library information system for the University) and external clients (notably as a member of the Plone.Gov project, which aims to introduce content management systems into 8 African parliaments). In addition, JKUAT has run training courses for Linux and e-government, through its commercial affiliate JKUAT Enterprise Ltd. JKUAT/SDU is building capacity for a range of technologies, and did not choose AVOIR software for the above projects.⁶

The need to focus more strongly business development around AVOIR software has been recognised by AVOIR. The initial work of building a base of software development capacity, is being complemented by efforts to

5 Email from Markus Pscheidt, Universidade Catolica de Mocambique, 2 October 2007.

6 Email Waweru Mwangi, 4 October 2007.

shift towards business development and a partnership with GeekCorps will bring short-term experts for this area to UWC and potentially other nodes.

Governance and Structure (Board level)

UWC has been very successful at marketing and promoting the AVOIR project and its outputs. It has established numerous partnerships with like-minded initiatives, attracted further funding, and presented the experience at conferences and workshops.

However, UWC finds itself in a difficult situation. On one hand its input and drive is required to animate the network, give direction to the software development, and mobilise additional sources of funding. On the other hand, by filling this important role, it allows the other network members to step back and rely on UWC. This imbalance is normal during a start-up phase, where one stronger partners supports others, but responsibilities need to shift away from one institutions to the network, for a sustainable community of practice to emerge.

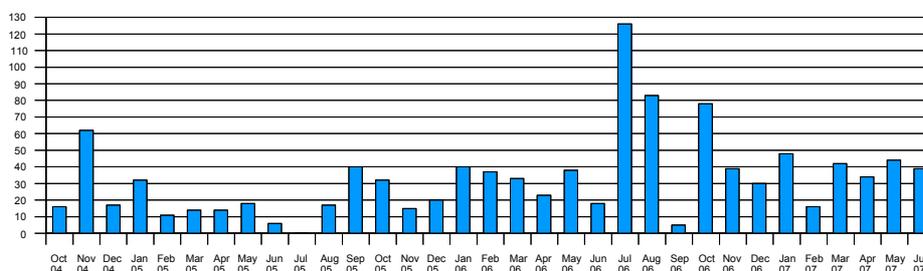
Main mechanisms for collaboration at the governance level were AVOIR Board meetings and communication via a mailing list.

Board meetings:

- Online meeting June 2006
- November 2006

Mailing list:

Illustration 4: AVOIR-board mailing list traffic in numbers of messages per month



There is relatively low traffic and activity on the list. Content analysis shows that traffic and discussion was driven by UWC and many of the active discussion threads are initiated by UWC – for example by announcing a new proposal and asking board members for comments.

Especially Prof. Derek Keats worked hard to stimulate debate, by introducing project ideas, draft proposals, and reminding board members of their responsibilities.

Participation and governance is discussed again in the context of software developers below.

Boundary partner: Software developers

FOSS developers in Africa face significant challenges. Africa lacks some of the infrastructure that is a requirement for participating in international FOSS projects. Africa does not have a long history and culture of Internet-use and bandwidth is still expensive and scarce. There are fewer FOSS developers and they are more distributed than in other parts of the world.

Outcome Challenge

The program intends to see developers collaborate actively in design and implementation of FOSS. Developers continuously work to expand and improve the software. Developers post questions, comments, and answer each others' posts on the mailing list. Experienced developers mentor less experienced developers, within an AVOIR node, and across nodes through use of email. AVOIR developers and ex-AVOIR developers are actively participating in other international FOSS projects, building on the skills they developed through AVOIR. Developers are comfortably using the tools and practices of international FOSS development. Developers support their institutions' efforts to achieve sustainability of AVOIR, by considering and addressing the needs of current and potential users of the software. AVOIR developers are promoting the use of FOSS and AVOIR software within their nodes, and in other organisations.

Developers who are not affiliated with AVOIR nodes

The evaluation team considered non-affiliated developers an important boundary partner, because a healthy free and open source software project is carried by a community of self-selected individuals. There are currently no developers that are not affiliated with AVOIR nodes or the AVOIR network.⁷ For this reason, we are not adding outcome challenges or progress markers for this group. This could be done at a later stage, as the project grows and is able to increase its community of developers.

Progress Markers

The program expects to see that developers are ...	Comments
... developing software development skills that are particularly relevant to FOSS	AVOIR has shown significant success in capacity building. See more detail below.
... designing and implementing software that addresses the needs of higher education institutions in Africa	The needs of higher education inform the development objectives of AVOIR software.
... soliciting user feedback and addressing it by developing new features and fixing reported bugs	While this has improved significantly over the past year or so, the project seems still somewhat driven by developers rather than user needs.
... participating on the developer mailing lists to report bugs and ask questions	The developer mailing list is very active. See discussion in software development section

⁷ This statement is based on the fact that AVOIR members have not provided any information on such developers, despite a number of requests for this data. The mailing list analysis (while not taking into account every single email) did not indicate that there are non-AVOIR developers.

	below.
... reviewing and improving software modules developed at UWC and sharing their improvements	Contributions from other nodes remain very low. See section on participation below.
... reviewing each others code, suggesting and supplying improvements	Based on mailing list communication there is little code review among developers between AVOIR nodes.
... using software tools like CVS comfortably and effectively	Few developers from AVOIR nodes report problems with CVS on the mailing lists, or mentioned it during interviews.
... developing skills and gaining experience that increase their potential for employment in their local economies	Participation in AVOIR increases job opportunities for developers. See further discussion under capacity building below.
The program would like to see that developers are ...	Comments:
... developing software that is equally or more appropriate for use in Africa than existing free and open source software solutions	It is difficult to assess if AVOIR software is better able to meet the needs of higher education in Africa than existing FOSS alternatives. Please refer to the software development section below for a longer discussion.
... observing the mailing list discussions and attempting to help other developers	Mailing list is very active and queries are usually addressed quickly. This indicates that a large part of the developer community does actively monitor the list.
... designing, implementing and contributing additional software code (new modules)	There is little contributed code from nodes except UWC. See further discussion under participation below
... showing a special effort to integrate and support new developers that are joining the network	The tone on the mailing list is typical for FOSS projects. It can be rough and discouraging to new developers, but the mailing list provides a good source of help and support for those able to adapt.
... testing each others code and reporting bugs and possible solutions	Judging from mailing list communication, developers usually install and try out new code, but do not test it in detail.
... documenting their code to make it easy for other developers to understand	Documentation has been a problem, which seems to be addressed after a recent push from UWC.
The program would love to see that developers are ...	Comments:
... connecting with local FOSS developers and integrating them into the AVOIR developer community	With the notable exceptions of Paul Scott and Prof. Derek Keats, there is little evidence that this is happening.
... working as project leaders or active participants in respected and successful international FOSS projects	With the exception of Paul Scott, AVOIR developers are not participating actively in other international FOSS projects.
... expanding the network by involving their new employers in AVOIR (this only applies to AVOIR developers that take new jobs outside of the University)	There is no evidence of this taking place.

Observations

Ownership

AVOIR provides an opportunity for young software engineers to learn about and experience FOSS development. In interviews almost all developers from AVOIR nodes mentioned that the project provided their first experience with FOSS and working in a distributed development project. One developer mentioned that participating in AVOIR helped him understand that money can be made from FOSS development. Developers were not aware of other FOSS development project of this scope in Africa; and especially none that provide resources (and funding) which they felt was crucial. AVOIR also provided an opportunity for developers to implement a project that fits with their belief that FOSS can provide an appropriate and effective technology solution for Africa universities.

Interviews with AVOIR developers showed high-levels of investment and a strong sense of ownership. Some developers mentioned that this kind of community was special in AVOIR and would be more difficult to achieve by simply trying to join an existing international FOSS project as a young developer.

Developers reported that they feel more ownership than they would if using proprietary or even other open source software. Developer mentioned that the barriers of joining these other FOSS communities are too high and that AVOIR lowers these barriers.

"I don't touch the base classes, but I can still understand how they work [by reviewing the code]. I can understand everything. That gives me a sense of ownership of the software."

AVOIR Developer (Uganda)

A strong sense of ownership seems to come from both the ability to understand *how everything works* by looking at the source code, and from contributing actively to the project. Developers stated that they would "only be users" if they were implementing and modifying other FOSS applications, whereas with AVOIR the developers really feel like they are part of the community that is responsible for the software.

"The AVOIR developers are like a community and I feel part of that community. I feel we are a family."

AVOIR Developer (Kenya)

Both the barriers to participation in other projects and the mechanisms through which AVOIR creates a sense of ownership (that developers do not expect to get from participating in other FOSS projects) need to be better understood.

Capacity building

AVOIR fills a crucial capacity building gap for software engineers. The main mechanisms through which the project builds skills and capacity are:⁸

⁸ Student projects (development as part of a degree programme) were also reported in the first AVOIR Technical Report, but seem to have been replaced by the success of the internship programme.

Developer workshops that bring together developers from all nodes

AVOIR experience has shown that bringing developers together face-to-face cannot be replaced by electronic communication.⁹

- January 2005, at UWC with developers from all nodes at the time (except one node, which sent an administrator)
- October 2005, at UWC with developers from all nodes
- November 2006, at UWC with developers from all nodes, and combined with a board meeting; 23 participants overall

Mentored internships of junior developers at some nodes

This is an initiative that was first introduced (and became very successful) at UWC, and is now being rolled out into other nodes. Each intern developer is linked with a senior mentor who can provide guidance, and support.

- During the past 2 years, at least 20 interns were trained at UWC
- At least 11 of these went on to jobs in the private sector after their internships¹⁰
- The Software Development Unit at Jomo Kenyatta University of Agriculture and Technology trained 3 interns who presented their work at an IEEE conference.

SDU has been a great place for our students on internship. 3 students have just finished their work there and all of them showcased their work at the recently held IEEE Centurium systems exhibition in Nairobi. One of them won a gift (a laptop). Waweru Mwangi, JKUAT, Email to AVOIR Board, 4 October 2007

Focused training workshops which take place at individual nodes

In order to strengthen nodes, and increase their ability to start contributing (and collaborate effectively using electronic communication) UWC developers visited a number of nodes for intense local programming workshops. In addition to developer training, UWC also provided e-learning implementation training to AVOIR nodes. AVOIR members also participated in, and spoke at international conferences.

2005:

- E-learning training workshop at Makerere University

2006:

- Developer training at Eduardo Mondlane University

2007:

- Developer and e-learning training at University of Dar es Salaam
- Developer and e-learning training in Kabul, Afghanistan
- Developer training at University of Jos

9 It was suggested to the AVOIR Researcher that analysing the effect of face-to-face meetings on mailing list traffic would provide useful indication of the value of these meetings. The analysis has not been provided to date, but could be added later to inform internal planning efforts.

10 Project Manager Abdul Fakier provided this information in his comments to an earlier draft of this document.

- Developer training at University of Ghana
- Developers from Kabul attending training in Cape Town

Ongoing knowledge sharing as part of the distributed software development process

Use of the mailing list to discuss the design of new features, solve problems collaboratively, and share solutions is the cornerstone of successful international FOSS projects. See below for further analysis of list traffic.

Distributed Masters Programme in Computer Science With FOSS Specialisation

Adding to the practical hands-on skill building, AVOIR nodes are currently designing a distributed Masters programme in Computer Science with a FOSS Specialisation. Courses will be shared between the institutions, and part of the instruction is provided online. The programme is currently going through an accreditation process at the partner institutions. It has been accepted in principle by Uganda Martyrs University.

As a result of this strong multi-faceted focus on capacity building, there is overwhelming agreement among developers that important skills are developed and that these increase their chances to find a job.

Both technical skills (database administration, etc.) as well as soft skills (project management, collaboration) were mentioned in interviews. With regards to technical skills the relevance for employment depends on the country though, since in some countries FOSS skills seem highly sought (Kenia, Uganda, Nigeria), whereas in others the situation is more difficult (Mozambique). A number of developers mentioned already being headhunted on the basis of their FOSS skills, either by other universities or the private sector. Another positive benefits is the reputation that comes from participating in an international project of the scope of AVOIR.

In addition, in many countries it is difficult for software developers to find work when they come straight out of University. AVOIR provides a great way to build relevant technical skills and real-life expertise to transition into employment.

"I am proud to be part of AVOIR. Being straight out of school it lets me utilise my skills."

AVOIR Developer (Kenya)

"AVOIR is a chance to be part of a professional international software development projects. This is an opportunity that did not exist for me before in my country."

AVOIR Developer (Mozambique)

"I learned a lot of new technologies and enjoy working with them. Before working on AVOIR, I would go to sleep early. Since joining AVOIR, I constantly think about coding and stay up late to be able to work."

AVOIR Developer (Nigeria)

The *downside* of the successful capacity-building activities are high turn-over since newly trained graduates quickly leave when offered better paid jobs. As developers gain relevant skills (see above) they get job opportunities. Universities need to find ways to either manage constant high turn-over or to budget appropriate salaries to retain staff.

Participation

While developers from many different nodes expressed a sense of ownership and collaboration, these sentiments are not reflected in mailing list postings and the amount of code that is committed to the AVOIR cvs server. The data shows that despite capacity building taking place, and the emerging sense of community, there is still a clear difference between the activity of UWC based developers (including international volunteers) and developers at other nodes.

The nextgen-online mailing list is the main form of communication between the developers. The list traffic has grown tremendously, the list now has more than 100 individual posters who generate over 800 messages per month.

Year	Avg. number of messages / month	Total number of posters
2004	339	34
2005	471	67
2006	665	110
2007	867 ¹¹	107

Table 1: Developer mailing list analysis¹²

However the vast majority of posts are contributed by a very small group of developers who are based at UWC. All top ten posters to the list up until December 2006 were based at UWC. During 2007 the top nine posters were based at UWC. The available data allows the estimation that only between 12 – 18% of messages on the developer list are posted by non-UWC developers.¹³

An even stronger position of UWC can be observed in terms of actual code developments, measured as number of commit actions to the central CVS server that holds all AVOIR software code. Numbers of commits are not necessarily representative of the amount of code that is contributed (a developer might work for a long time and then commit a large amount of new code). However, given the significant difference between UWC and other partners it is unlikely that one of the partners is indeed a significant contributor.

11 During 2007, UWC started making active use of the mantis bug tracking software which generates a significant amount of mailing list traffic. The mantis contribution was controlled for – the number reported here represents only direct email contributions.

12 Raw data was provided by AVOIR Researcher Enver Ravat.

13 Based on own analysis of basic data provided by AVOIR Researcher Enver Ravat. See Annex for more detail.

Contributor	Number of Commits to CVS
UWC	4259
GeekCorp ¹⁴	68
JKUAT	52
Makerere	7
Jos	2
Others	0

Table 2: CVS commit statistics¹⁵

It is important to note that UWC made significant efforts to encourage other nodes' contributions, and support their capacity-building. Messages on the board mailing list shows repeated reminders from UWC to the other board members to increase their activities.

“Despite the workshop in November and everything we agreed, there is almost no code coming into CVS (except for JKUAT and U. Nairobi), and only JKUAT is making use of the mailing list. I am aware that a few are busy with translations, and that doesn't give much to talk about, but complete silence? UWC people account for 99% of the mailing list traffic, and David Wafula and Ryan Sain (Washington State) the rest. This is not what we agreed.”

Prof. Derek Keats, AVOIR-Board mailing list, 1 March 2007:

The following statement from the first AVOIR Technical Report shows that AVOIR was aware of this problem, and as a result started implementing more on-site developer workshops, in which senior UWC developers spent time at partner institutions to provide hands-on training. While the results in terms of contributed code have not changed significantly yet, it might simply take more time for the training to pay off – more face-to-face workshops seem a promising strategy.

“This workshop introduced the developers to the KINKY application framework, the procedures for communication among developers, our approach to object-oriented development, the structure of KEWL.NextGen modules, and version control using CVS. ... While some of the partner developers understood the approach and have contributed code to the project, it seems as though the contact period was too short for some of the less experienced developers. Hence some developers are struggling to make contributions.”

African Virtual Open Initiatives and Resources Project, Annual report for 2005, Prof. Derek Keats & Paul Scott

User focus

One characteristic of FOSS projects (and one reason for their success) is that they are inherently driven by developers. This is also the case for AVOIR software. However, this developer focus can become a challenge if communication between users and developers is not effective.

¹⁴ GeekCorp refers to volunteers who were based at UWC

¹⁵ Source: Email, Prof. Derek Keats to nextgen-online discussion list, May 30 2007

AVOIR provides a mailing list for users of the software, which has relatively low traffic, but provides a good way of finding support or comment. Based on personal experience of the evaluators, requests are usually responded to promptly and the senior developers of the project make efforts to understand and address their users.

Documentation of the software has been a problem in the past, but a recent push by UWC has significantly improved the situation. It is now much easier to find installation instructions on the AVOIR site, and there has been strong emphasis on documentation of the software code on the developers mailing list.

The possibility for users to inform which new features are created and how they are implemented is limited. As discussed in other parts of this document, prioritisation of tasks and feature requests is based on (paying customers' needs first, and then) developer interests rather than what users perceive as their needs. See more on this in the next section for the perspective of AVOIR affiliated institutional users.

Boundary partner: Users of the software developed by AVOIR

We are using the e-learning team at UWC as an example user of the software for a number of reasons. First, UWC is the most active user of the software developed by AVOIR and has been over some time. Second, the e-learning team has recently conducted an internal review from which we can draw information. However, looking at one user in such detail also means that the findings are not representative – many things differ between universities. We are trying to focus on those issues that are most likely to be applicable to other users, and leave out some of the very UWC-specific experiences.

Outcome Challenge

Note: The outcome challenge for a software user is intrinsically linked with the quality of the software produced by the AVOIR developers, since users have limited means to influence the development of the software.

The program intends to see users of its software act as active contributors to the development and testing process of software developed by AVOIR. In the case of e-learning, users of KEWL (the AVOIR learning management software) will promote the concepts of e-learning and the use of KEWL on campus. E-learning teams raise awareness and provide training and ongoing support to lecturers and students. They collect their feedback and present it to the developer community by posting it to the mailing list and meeting with their local AVOIR node developers (where available). E-learning teams conduct and present research on e-learning and the use of KEWL at conferences and to potential other users and clients.

Progress Markers

The program expects to see that e-learning users are ...	Comments:
... participating in the design and testing of new version of the learning management software	The e-learning team at UWC has been actively involved in the design, testing and bug reporting
... contributing user feedback based on their experience with lecturers and students	Again, the e-learning team at UWC has done an excellent job reporting user experience in order to improve the software.
The program would like to see that e-learning users are ...	Comments:
... designing new features of how KEWL could be expanded based on their experience with lecturers and students	Designing and requesting new features has taken place, but the implementation has not been realised as desired.
The program would love to see that e-learning users are ...	Comments:
... making a measurable contribution to the quality of learning and teaching (education) at their institutions	This is definitely the case at UWC, and can be partly attributed to AVOIR.

Observations

The positive change that AVOIR is bringing about is related not only to the number of developers that are involved, but also to the number of people that use its software, and their satisfaction with it (which indicates how well it supports their work).

Popularity of the software

Since anyone can download and install the software there are no reliable ways of determining the exact number of users. Project Manager Abdul Fakier estimates that there are more than 40 production sites using AVOIR software.¹⁶ In addition, many more seem to be installing the software for personal use. For example, packaged installer version of the learning management system KEWL.NextGen were downloaded just under 800 times in August 2007.¹⁷

As an indication of more general interest in the project (not just its software), one can consider the number of page views. In August 2007, there were roughly 81,472 page views.¹⁸

¹⁶ In a comment to an earlier draft version of this document.

¹⁷ http://avoir.uwc.ac.za/webalizer/usage_200708.html

¹⁸ The total number of recorded page views was 193384, but at least 57.87% came from search engine bots; http://avoir.uwc.ac.za/webalizer/usage_200708.html; unfortunately there is no comprehensive analysis of web traffic for the relevant sites over the evaluated period.

User satisfaction

The defined progress markers only focus on the work of the e-learning team. However, to a large degree the e-learning team's ability depends on responsiveness and service they receive from the local AVOIR developers working on KEWL. User satisfaction ranges from excellent (see quote from Ryan Sain below) to mixed in the case of the UWC e-learning team.

Some of the key challenges that were reported by the e-learning team include lack of responsiveness to feature requests and bug reporting; insufficient testing of the application before implementation; lack of benchmarking; a developer focused attitude that does not consider users the most important stakeholders in this project; and limited focus on design and usability – which the e-learning team considers crucial for success.

For example, requests to improve the look-and-feel of the application in order to support an important international collaboration with a funder were not addressed. Yet, one lecturer's request for additional features was implemented, mainly because of the lecturer's persistence.

While these challenges can be considered necessary drawbacks that are balanced by the capacity-building success of AVOIR, users outside of AVOIR will evaluate the software purely on its merits.

The challenges reported by the e-learning team have created frustration among users, and might prevent some of the benefits that are generally expected from using ICTs effectively for learning to materialise.

“Just an update about a recent success we had with KNG.

We have adopted KNG as the elearning system for higher education in Afghanistan. This initially started at Kabul University and has spread to Kabul Polytechnic University and soon to Kabul Medical University.

Long story but the big point is this: one professor at Kabul Polytechnic University is actually using the system to support/teach his current face to face course. This is a first in Afghanistan. He stated that the reasons he likes it is because the resources are always available to his students and it makes marking exams easy and transparent and fair. These are big issues here in Afghanistan. So much of a big issue that the media got involved. Local TV stations came out and interviewed him and some of our team about eLearning in Afghanistan. This is the second step toward critical mass! We will be getting the video soon and will be posting it to our CMS site.

This would not have been possible without the years of tireless efforts by the AVOIR teams to get this functional and stable.

Thank you for all your work.”

Ryan Sain, Ph.D., Deputy Chief of Party, Afghan eQuality Alliances

Additional boundary partners

We did not have access to data that would allow observations of behavioural changes in a number of boundary partners, which we initially intended to include in this evaluation.

- Clients of services and products offered by AVOIR nodes
- University management and administration

We have added our draft text for outcome challenges and progress markers for these boundary partners in the Annex and hope they can be included in future monitoring and evaluation efforts.

Software development

The OM framework is not intended for analysis of software development processes. The software development choices and processes are described through a combined review of mailing list activity, software development indicators (lines of code, numbers of contributors, etc.), comparison with other FOSS projects, and contain a brief review of the competitive environment for AVOIR software.

Design – Framework and applications

AVOIR has designed and built a web–application–development framework, and created applications that are based on this framework.

The framework is called Chisimba (an earlier version was called KINKY) and is implemented in the PHP scripting language. It requires a database and webserver. Some of the applications that were created using the framework are KEWL (or KewlNextGen, a learning management system), Kgroups (a discussion forum application), and recently a blog. In addition, a huge number of modules were created to be combined to create new applications within the Chisimba framework.

AVOIR software competes with existing open source and proprietary products in a number of markets, including application–development–frameworks, e–learning systems, blogs, etc.

Development processes

Large–scale and distributed FOSS development processes pose complex management challenges. For one, a balance must be found between each developer's ability to work on tasks that is of interest to them (Eric Raymond refers to this as “scratching an itch”) and the overall deliverables that are expected by clients and users of the software.

AVOIR has designed an elaborate software development process based on Agile programming methodology

“Our approach to agile methods attempts to minimize risk by developing software in short time periods, called iterations, which typically last four weeks or less. Each iteration is like a miniature software project of its own, and includes all the tasks necessary to release the mini–increment of new functionality: planning, requirements analysis, design, coding, testing, and documentation ().”

African Virtual Open Initiatives and Resources Project, Annual report for 2005, Prof. Derek Keats & Paul Scott

However, as in most development projects, these well designed processes are often adjusted to deal with urgent requests, troubleshooting, and to meet deadlines for client projects.

Over the course of the project a number of changes and improvements have been made, which take into account the looseness and flexibility of FOSS development, and the need for some structure and accountability, including:

- Tracking of bug and feature requests via the MANTIS application, which allows assigning tasks to developers and stores all progress information in one central place.
- Setting up a system for release of packaged applications rather than requiring users to access the development code that is stored in the Concurrent Versioning System (CVS) repository.

One area for improvement is prioritisation of development tasks. Feature requests, and bugs are reported by users and enter a queue. The basic rule at UWC is that paid projects take precedence, so new features are implemented first for those who are willing to pay for them. That makes sense. It is less clear how the long list of other tasks is prioritised, and to some degree the developers remain free to pick what tasks they are working on. Developer interests tend to focus on adding new features, rather than simplifying the application and improving usability.

Competitive environment

Application–Development–Frameworks

Chisimba has about 3 times more active developers, and 8 times more lines of code than the other application–development–framework projects.

Project	Type of software app	Lines of Code	Active developers
Chisimba	Application Development Framework	687703	35
Symfony	Application Development Framework	89750	7
Ruby On Rails	Application Development Framework	86141	13
CakePHP	Application Development Framework	50794	7
Moodle	Learning Management System	772565	81
Moodle add-ons		502218	44

Table 3: Lines of code and Active developers¹⁹

Given that some of these projects are very widely used and popular, this is surprising, but can be explained by the approach that AVOIR is taking in the design of Chisimba. The Chisimba project combines the basic framework with a lot of application modules. The code of the modules is included in the above statistics. While the benefit from modular architecture comes from the independence of modules from each other, the combination of framework with modules (and between modules) in Chisimba is structurally embedded in the implementation of the software.

¹⁹ All data collected during May / June 2007 from <http://opencollabnet.ohloh.net>

“If you try deploy just Chisimba without the modules you run into all kinds of issues where modules depend on other modules which in turn depend on other modules. I'm not sure developers have paid enough attention to keeping their modules "modular" whenever possible.”

Adrian Woodhead, nextgen-online mailing list, 4 June 2007

Other frameworks provide *hooks*, that enable integration of other applications to provide certain features. For example, CakePHP has modules that allow it to integrate with CMS applications like Joomla and Drupal, rather than provide a CMS itself. The strategy of Chisimba so far has been to provide all desired features and functionality through native Chisimba modules, even if these modules are not as stable and feature-rich as existing open source alternatives (discussions are underway on how to provide a more standardised interface for combination with modules from other providers).

Recently new applications have been developed that embrace the web as a platform, tying together different sources of functionality and data, and blurring the boundaries between content producers and users. This has been described as Web 2.0 by some.²⁰ Chisimba, while tying into some Web 2.0 services (for example, videos from youtube.com can be included in some modules of Chisimba applications) aims to provide most functionality within one integrated application framework through modules specifically developed for this framework (and by AVOIR developers). AVOIR developers point out that Chisimba can be integrated in Web 2.0 applications, because it offers an open Application Programming Interface that others can write software for, and that creating an integrated solutions also has important benefits:

“But, there are perfectly good reasons to have an integrated system, even if there is less functionality. For example, having tagging that runs across all content. Go to photogallery and send this picture to my blog. Add the content of this blog post to my e-portfolio. etc. That's what we must be working towards. If all we are doing is gluing together unintegrated functionality, then we really have no competitive edge.”

Prof. Derek Keats, Email to nextgen-online mailing list

This is an area where opinions of the evaluator and some AVOIR developers diverge, and a more detailed technical review of the AVOIR software design in comparison with other frameworks and Web 2.0 practices could provide further insight.

Learning management systems

Compared with other popular learning management systems, the sheer mass of users and developers of other open source solutions creates a fiercely competitive environment. In a recent report by the elearning guild for which over 840 institutions were surveyed, Moodle received the highest user satisfaction of any LMS in the education and government learning management sector (beating out proprietary competitors).²¹ The opencollabnet data shows that the Moodle community has at least twice (and possibly up to three times if counting both core and module

20 See http://en.wikipedia.org/wiki/Web_2

21 <http://www.elearningguild.com/pbuild/linkbuilder.cfm?selection=doc.1373>

developers) the amount of active developers, and that they have produced significantly more lines of code. A closer look at individual commit statistics also shows that the level of activity is more widely distributed, whereas for Chisimba a small group of developers is responsible for virtually all commits.

Other applications (blogs, discussion forums, wikis, etc.)

Widely used open source software alternatives exist in most other areas that AVOIR competes in. For example, wordpress is a popular open source blog application used on “hundreds of thousands of sites”²², phpBB provides community-building and discussion forum features and is installed on “millions of sites”²³, and there are a multitude of popular wiki applications to choose from²⁴.

Conclusions and Recommendations

AVOIR has been a tremendous success as a capacity- and network-building project in the free and open source software sector in Africa. Through workshops, training programmes, and online collaboration the project has created a strong spirit of ownership and developers report that they feel part of the “AVOIR community”. AVOIR has a unique approach to capacity-building, which is helping fledging FOSS developers develop the skills they need to start collaborating on international FOSS projects, and/or find employment locally. Many interns and young developers have been offered jobs as a result of their involvement in AVOIR. However, AVOIR also struggles with the tension between successful capacity-building and developing and marketing “world-class” software. Its software is not as widely used as desired, some of AVOIR’s key users are not satisfied with the design and reliability of the software, and only UWC has been able to develop a FOSS business model based on services and support contracts for implementation of AVOIR software. Despite the absence of a formal monitoring and evaluation system, AVOIR has been very good at identifying challenges, and developing new solutions to address them.

“One last factor: Developers here are not capable to jump start into proper software development. Actually, the goal that rather can be targeted in a short time perspective (months) is that new developers learn writing software by learning from source code that is present in KEWL. This means a certain level of capacity building is the big gain for a university like us. I know, that doesn't help the development of the KEWL software itself.”

Markus Pscheidt, AVOIR-Board mailing list, 1 March 2007

22 <http://wordpress.org/about/>

23 <http://www.phpbb.com/>

24 <http://www.wikimatrix.org/>

Challenges and recommendations

Business development

With the exception of UWC, most AVOIR nodes have not engaged in significant business development or been able to create sustainable local FOSS ecosystems. A piece of the FOSS ecosystem is missing: connecting development of the software, to the provision of marketing and support services for it to clients. However, it has already been acknowledged that a stronger focus on this area is required to ensure long-term sustainability. Future evaluations will be able to assess the results of recent initiatives launched to support the nodes in marketing FOSS products and services.

Software

AVOIR faces the difficult task of developing "world-class" software while building capacity in Africa. It has to invest a lot of energy in keeping the development process of a large and complex application going; energy that cannot be invested in building business models or increasing capacity-building efforts. AVOIR is not developing completely new types of applications, which did not exist before and there is strong competition for AVOIR software. While AVOIR software has not seen the level of uptake and use that was desired, especially outside of the AVOIR community, it is too early to rule out that this will happen. However, user dissatisfaction with AVOIR software is significant and needs to be taken very seriously to increase competitiveness with other existing applications.

Governance

UWC finds itself in a difficult situation. On one hand its input and drive are required to animate the network, and guide software development, and develop new proposals. On the other hand, the network relies too strongly on UWC playing this role. Distributing governance and responsibility more evenly is one challenge that has to be addressed to enable the formation of a truly sustainable community of practice.

Recommendations

- Redefine success as broad capacity development – AVOIR should build on the success it has had as a capacity-building project; extend its intern programme and add additional skills (other technologies, business skills, etc.) to it. There is currently no strategy that considers how AVOIR could be successful as a capacity-building project if Chisimba software does not find the desired uptake and success; in other words, AVOIR success is tied to the success of Chisimba software. One consequence is that all capacity-development efforts are focused exclusively on Chisimba software. While some secondary skills are developed as part of this process, applying Chisimba expertise in the context of other more widely used applications (drupal content management system, wordpress blog, moodle learning management system), would

require further training. Another consequence is that it has so far limited how AVOIR capacity-building can tie into other FOSS capacity building projects that focus on non-Chisimba software (for example, the Google Summer of Code, or inWent's support for generic FOSS skills). While the AVOIR Executive Director considers the exclusive focus on Chisimba an important requirement that will ultimately allow AVOIR to succeed, the evaluator would recommend a more diversified approach that builds capacity for a variety of applications relevant for African higher education, including the most widely used (best of breed) ones in their fields. Diversification would shift the focus on broad capacity building and business development and consider software development (for a range of applications) as a means to these ends.

- Push responsibility into the network – To become sustainable as a distributed community of practice, AVOIR needs to find ways to increase involvement and activity among its nodes, and reducing the dependence on UWC's vision and leadership. Unfortunately this is not a trivial process. Reviewing the current list of nodes and creating different levels of membership based on contribution (for example, only active members are on the board, or get funding support from the network) could be a first step.
- Find a niche and exploit it – Competing in all areas (wikis, blogs, discussion forum, e-learning applications, etc.) is a daunting task for a small project like AVOIR. The evaluator recommends that chisimba either be slimmed down to be *just* a framework and one showcase application in an attractive niche be developed on top of it; or that AVOIR define a portfolio of the most widely used FOSS applications (mediawiki, wordpress blog, etc.) and add modifications/ extensions to these applications in order to address specific needs of African universities. The AVOIR Executive Director disagrees with the evaluation on this point. Prof. Keats feels strongly that the focus on an integrated suite of modules is not a challenge, but will instead be one of the reasons leading to widespread uptake and success of AVOIR software.
- Make it easy to join – AVOIR's communication strategy should be clarified and nodes supported in their efforts to reach out to local communities. The low level of outside involvement (most participants are tied to AVOIR partners in some way) could be related to the lack of step-by-step instructions on how to join the project. The website should be used more extensively to publish community news and always offer up-to-date quality documentation. Getting developers to participate in PR and marketing is a challenge and might require some form of incentive, for example the author of the most active blog is funded to attend an international conference.
- Provide solutions, not software – Similar to FOSS developers in Africa, FOSS users require more support than users in other parts of the world. Appropriate solutions therefore include not only software, but packaged software and support and training services.

The business component of AVOIR could identify key user needs, designs solutions to address these needs, and market them.

- Care about your users – Thinking like a user ensures that outputs address real needs. A stronger focus on users implies benchmarking against other solutions, and a shift of development priorities from adding new features, to improving the user–interface and usability of software. Good development practice would include careful prototyping and testing stages. A specific process improvement needed is clear prioritisation of development that balances both urgent needs (problem fixes), developer interests, and strategic objectives with user feedback. Working with international volunteers has been a good way of introducing new processes and tools, and this practice should be expanded. Staff exchange between nodes could then disseminate best practices and new processes from one node to the others.
- Increase monitoring, research and reflection – Strengthen research activities in order to:
 - (1) Conduct ongoing monitoring and evaluation of the project success and challenges. For example, the project should keep track of the interns and developers that gain skills and move on to other jobs; and record and regularly analyse web–site and software download statistics.
 - (2) Fill a crucial gap in the research of how open collaborative processes can best be implemented in a developing country environment. The AVOIR experience provides a fascinating source of data to further analyse and learn from.
- Leverage the existing network – AVOIR has been able to create a network of software developers (with varying degrees of activity across nodes). The network should consider if there are related activities that it could implement. These do not necessarily involve the development of software (for example, moving into open content, and collaborative instruction).

Annex A Outcome Mapping Overview

The following text was copied from: Brochure on Outcome Mapping: The Challenges of Assessing Development Impacts Sarah Earl, Fred Carden, Terry Smutylo and forward by Michael Quinn Patton, October 2001. Available at http://www.idrc.ca/en/ev-62234-201-1-DO_TOPIC.html

Outcome Mapping focuses on one particular category of results – changes in the behaviour of people, groups, and organizations with whom a program works directly. These changes are called "outcomes." Through Outcome Mapping, development programs can claim contributions to the achievement of outcomes rather than claiming the achievement of development impacts. Although these outcomes, in turn, enhance the possibility of development impacts, the relationship is not necessarily one of direct cause and effect. Instead of attempting to measure the impact of the program's partners on development, Outcome Mapping concentrates on monitoring and evaluating its results in terms of the influence of the program on the roles these partners play in development.

In the IDRC context, defining outcomes as "changes in behaviour" emphasizes that, to be effective, development research programs must go further than information creation and dissemination; they must actively engage development actors in the adaptation and application. Such engagement means that partners will derive benefit and credit for fulfilling their development roles whereas development programs will be credited with their contributions to this process. With Outcome Mapping, programs identify the partners with whom they will work and then devise strategies to help equip their partners with the tools, techniques, and resources to contribute to the development process. Focusing monitoring and evaluation on changes in partners also illustrates that, although a program can influence the achievement of outcomes, it cannot control them because ultimate responsibility for change rests with its boundary partners, and their partners and other actors. The desired changes are not prescribed by the development program; rather, Outcome Mapping provides a framework and vocabulary for understanding the changes and for assessing efforts aimed at contributing to them.

Outcome Mapping

- Defines the program's outcomes as changes in the behaviour of direct partners
- Focuses on how programs facilitate change rather than how they control or cause change
- Recognizes the complexity of development processes together with the contexts in which they occur
- Looks at the logical links between interventions and outcomes, rather than trying to attribute results to any particular intervention
- Locates a program's goals within the context of larger development challenges beyond the reach of the program to encourage and guide the innovation and risk-taking necessary

- Requires the involvement of program staff and partners throughout the planning, monitoring, and evaluation stages

Terminology

Boundary Partners: Those individuals, groups, and organizations with whom the program interacts directly to effect change and with whom the program can anticipate some opportunities for influence.

Outcomes: Changes in relationships, activities, actions, or behaviours of boundary partners that can be logically linked to a program's activities although they are not necessarily directly caused by it. These changes are aimed at contributing to specific aspects of human and ecological well-being by providing the boundary partners with new tools, techniques, and resources to contribute to the development process.

Progress Markers: A set of graduated indicators of changed behaviours for a boundary partner that focus on depth or quality of change.

Annex B Developer mailing list analysis

The developer mailing list (nextgen-online) archives are at:

<http://avoir2.uwc.ac.za/pipermail/nextgen-online/>

The basic data for this analysis was provided by AVOIR researcher Enver Ravat. Analysis was done by Philipp Schmidt.

Overall traffic per month and year:

Period	Avg / month	No. of messages	No. of messages	Total / year
2004 (May-Dec)	339.25	2714		2714
2005	471.5	5658		5658
2006	665.42	7090	895	7985
2007 (Jan-June, incl. Mantis)	1306.33		7838	7838
2007 (Jan-June, excl. Mantis)	867.67		5206	5206

Eliminating the effect of Mantis

Messages from: mantis@uwc.ac.za	2349
Messages from: mantis@avoir.uwc.a	283
	2632

Traffic from UWC:

Period:	Sender domain:	Comment:	No. of Messages	% of all msgs.
2004 – 2006	uwc.ac.za	(developers)	12216	
	gmail.com	(est. 20% of this P. Schmidt)	588	
	datamage.net	(Sean)	470	
		Total msgs from uwc	12803.6	
		Total no. of msgs	15462	
		Total msgs from others	2658.4	17.19

Period:	Sender domain:	Comment:	No. of Messages	% of all msgs.
Dec 2006 – June 2007	uwc.ac.za	(developers and mantis)	6606	
	gmail.com	(est 80% Adrian Waldhead and Philipp Schmidt)	582	
	greenlikeme	(Ryan Whitney)	411	
	avoir.uwc.ac.za	(mantis)	283	
		Total msgs from uwc	7765.6	
		Total no. of msgs	8733	
		Total msgs from others	967.4	11.08

Annex C Additional boundary partners

***NOTE:** We did not have access to data that would allow observations of behavioural changes in some boundary partners, who we considered useful.*

- *Clients of services and products offered by AVOIR nodes*
- *University management and administration*

We have included draft outcome challenge text and progress markers for these boundary partners in the Annex and hope they can be included in future monitoring and evaluation.

Clients of services and products offered by AVOIR nodes

AVOIR is aiming to create an ecosystem of FOSS development in which the local nodes are able to provide services within their institutions as well as to clients in the local economies. The income from these services can then support the software development and capacity building activities at the nodes.

However, many firms are long-time users of proprietary software and lack awareness of FOSS solutions. Business development in the FOSS sector in Africa is time- and resource intensive and requires special business and marketing skills that do not typically exist in universities.

Outcome Challenge

The program intends to see that organisations shift from using proprietary software to open source software by evaluating and implementing FOSS solutions in collaboration with local service providers. Organisations are considering to move from being consumers of software products, to participation in the production and marketing as part of a FOSS ecosystem.

Progress Markers

The program expects to see that external organisations are ...

- considering and evaluating locally developed FOSS solutions to supplement existing proprietary systems

The program would like to see that external organisations are ...

- open to participate in research surveys on FOSS
- contracting AVOIR nodes for design, development, implementation or maintenance of FOSS solutions
- participate as “users” on the mailing list

The program would love to see that external organisations are ...

- willing to participate in collaborative open source projects and engage in the AVOIR mailing lists

- promoting AVOIR developed software to others
- report bugs, and make suggestions for improvements

University management and administration

As clients (Nursing department, HR department, Centre for the Study of Higher Education (CSHE), ...

As supporters (Rector to promote the ideas of knowledge sharing, fostering relationships for the University)

Outcome Challenge

The AVOIR program intends to see UWC management and administration fully engage with the UWC node of the AVOIR project. UWC management regularly meets with the local AVOIR node to discuss progress and collaboration. UWC management collaborates in the development of AVOIR software by providing regular feedback to the developers. UWC management and administration have evaluated and deployed AVOIR developed software to support the day-to-day operation of the university, successfully replacing proprietary systems. Senior management at UWC promotes the benefits and uses of AVOIR software inside and outside of the institution. This include efforts to raise awareness, attract funding, and build relationships with other organisations. The institution has created an environment that makes best use of ICTs in education, students have access to computers, faculty and staff get training and support, and software is developed to address the needs of UWC.

Progress Markers

The program expects to see that UWC management and administration are ...

- considering implementation of AVOIR developed software within the institution for management and administration services (and contracts the services of UWC's AVOIR node)
- promoting AVOIR software to the departments of UWC
- meeting regularly with AVOIR representation to discuss strategic collaboration

The program would like to see that UWC management and administration are ...

- promoting the use of AVOIR developed software outside UWC, to other universities and the private sector – and contributing to sustainability of the AVOIR node
- creating an environment that enables best use of AVOIR developed software to support the various activities of UWC's campus community

The program would love to see that UWC management and administration are ...

- suggesting new features to improve AVOIR software