Free & Open Source Software Conference 2015

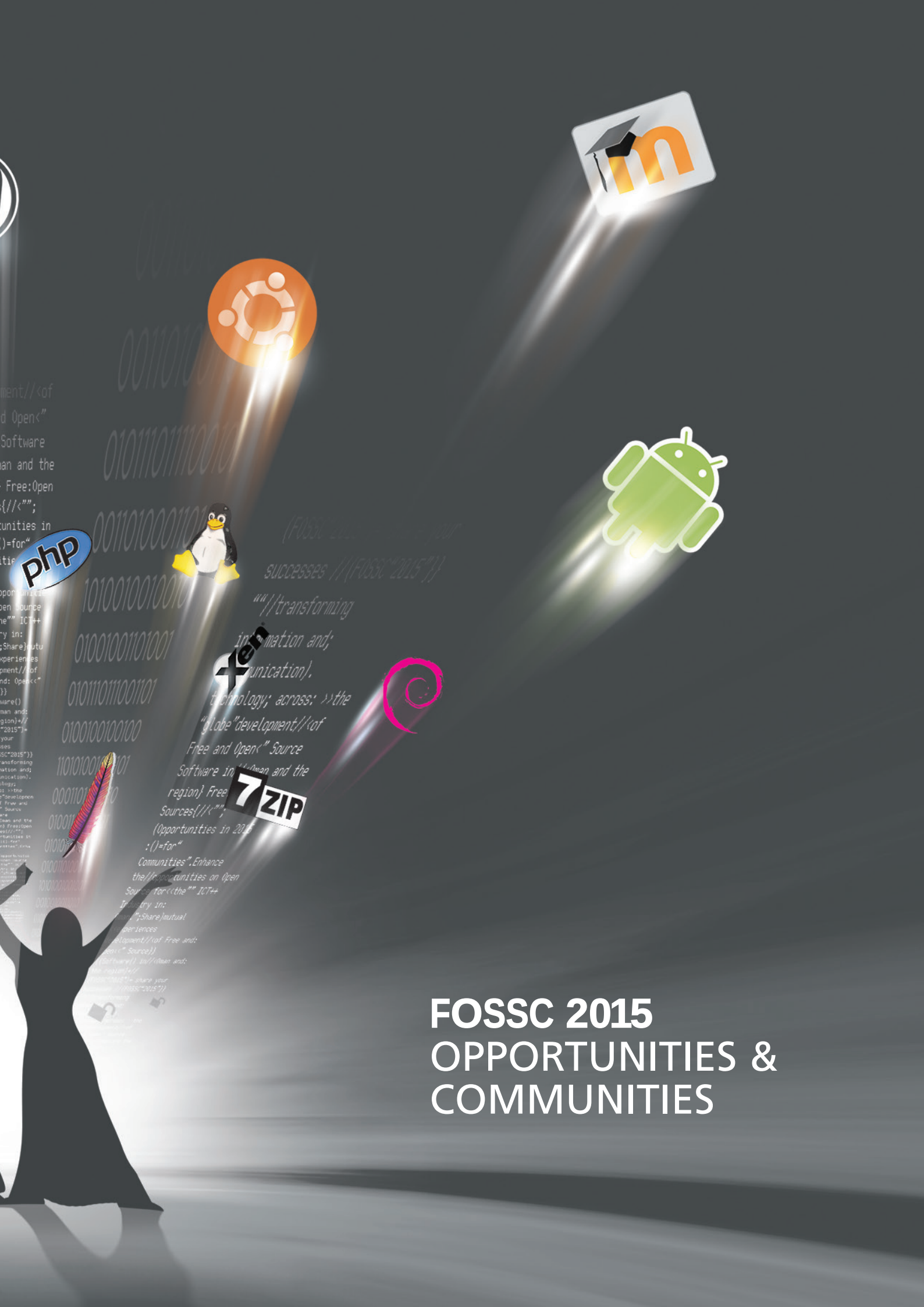
PROCEEDINGS

Sultanate of Oman
Information Technology Authority

جامعة السلطان قابوس
Sultan Qaboos University

e.oman


development//<of Free and
Open<" Source Software
in//<Oman and the region}
Free:Open Sources{//<"";
(Opportunities in 2015
:()=for"
Communities".Enhance
the//opportunities on
Open Source for<<the""
ICT++ Industry in:
Oman.";
<<exper
developm
and: Open
((Software() in
and: the region}+//
(FOSSC"2015")+ share your
successes
//(FOSSC"2015"}}
""//transforming
information and;
(communication).
technology: across:
"globe"de
Free and
Software


FOSSC 2015
OPPORTUNITIES &
COMMUNITIES

PROCEEDINGS

FREE & OPEN SOURCE SOFTWARE CONFERENCE

FOSSC-OMANí2015

ìFOSS Opp ortunities & Communitiesî

Organized jointly by SQU and ITA

Held at Sultan Qaboos University on 18-19 FEBRUARY, 2015.

Sponsors

Supporters

Organizers

Sultanate of Oman
Information Technology Authority

Free and Open Source Software Conference Oman 2015

The Free and Open Source Software Conference, FOSSC-Oman'2015 is jointly organized by the Communication and Information Research Center at Sultan Qaboos University (SQU), and the Information Technology Authority (ITA). The Main goal is to support the National Initiative on dissemination and development of FOSS solutions in the public and private sectors in the Sultanate. The event is scheduled for two days and targets ICT professionals, academic and technical staff, students, and the broad community in Oman to contribute in promoting awareness, dissemination and development of free and open source software.

The FOSSC-Oman'2015 is the second edition of the FOSSC-Oman conference held every two years in Oman. This will offer a good opportunity to ICT managers, professionals, developers, users and interested persons from Oman and elsewhere to discover, discuss and exchange views on FOSS solutions and challenges. The event will also provide an opportunity to network different groups of users, benefit from insightful lectures delivered at the conference, and share mutual experiences and implementations of FOSS products.

The organizing committee is inviting all of these individuals and groups to share their achievements in the vast array of free and open source software which is transforming ICT domain across the globe.

Communication and Information Research Center (CIRC)
Sultan Qaboos University
P.O. Box 50
Postal code 123
Al-Khoud, Muscat,
Sultanate of Oma

Table Of Contents

Message from the Organizing Committee

The organizing committee has a great pleasure to welcome you to the second edition of FOSSC-Oman Conference on Free and Open Source Software, which is jointly organized by the Communication and Information Research Center at Sultan Qaboos University (SQU), and the Information Technology Authority (ITA).
The second FOSSC-Oman edition 2015 main theme is concentrated on FOSS Opportunities and Communities. Within this line, the organizing committee has invited a number of well-known experts who are actively involved in building careers, entrepreneurship and job creation, sustainability of open source communities, role and opportunities of FOSS in public administrations, open source for the public sector, scaling applications for global communities, business management, business information systems, e-collaboration, educational Management and other related subjects.

Besides this, a good number of papers on local and international experiences on FOSS deployment in academia and other sectors have been selected by the scientific committee. The program committee has scheduled seven workshops for the community to be presented in parallel sessions.

The conference organizing team is greatly grateful to all members of various committees from SQU and ITA, and to the reviewers for their worthwhile efforts, time spent and quality assistance to make FOSSC-Oman'2015 a useful addition to their activities.

We would like to wish you all success in your scientific and technical presentations, discussions and social networking. We also hope that you enjoy your stay in Oman and we look forward to meeting you in the near future.

Prof. Hadj Bourdoucen, Chair
Organizing and Scientific Committee.

Executive Committee

Conference Chairs
H.E.Dr. Ali bin Saud Al-Bemani,
Vice Chancellor, Sultan Qaboos University (SQU).
Dr. Salim Sultan Al Ruzaiqi,
Chief Executive officer of Information Technology Authority.

Organizing Committee Chairs
- Hadj. Bourdoucen, SQU
- Fahad Al Abri, ITA

Advisory Committee
- Amer A. Al Rawas, DVC PSR (Chair)
- Saif Al Haddabi, TRC
- Hadj Bourdoucen, CIRC
- Fahad Al Abri, ITA
- Ali Al Shidhani, TRC
- Khaled Day, SQU
- Ahmed Al Naamany, MCBS
- Mohamed Sarrab, CIRC
- Hassan Al Lawati, ITA
- Fahad Al Saidi, ITA
- Khalil Al Maawali, ITA

Scientific Committee
- Hadj. Bourdoucen, (Chair)
- Fahad. Al Saidi
- Khaled. Day
- Mohammed. Ould Khaoua
- Fahad. Betishiginah,
- Dawood. Al Abri
- Mohamed Sarrab
- Zuhoor. Al-Khanjari
- Nasser. Al Zidi
- Amer. Al Habsi
- Hafedh. Al Shihhi
- Ahmed. Al Maashri
- Ashraf. I. Saleem
- Khalil Al Maawali
- Mehdi. A. Kordestani

Publicity Committee

- Samia. Loucif
- Mehdi. A. Kordestani
- Sumaiya AlKindi, ITA
- Osama. Rehman
- Said. Al-Muharrami
- Muadh. AL-Hinai
- Ibrahim Al Zakwani

Organizing Committee

- Ahmed Al Busafi
- Ali Al Adawi
- Fahad Al Saidi, ITA
- Khalil Al Maawali, ITA
- Ibrahim Al Wardi, ITA
- Tofool Al Dhahab, ITA
- Asad Al Adawi, ITA
- Abdullah Al Dawoodi, ITA
- Jokha Al Shabibi, ITA
- Adeeb Al Mendhry, ITA
- Said. Al-Muharrami
- Mohammed Al Ghailani
- Al hassan Al sobhi
- Hassan Al Hosni
- Rashid Al Ouhibi
- Ahmed. Al Farsi
- Ahood. Darmaki
- Younis. Al Kharusi
- Halima Al-harrasi
- Asharul Khan
- Shaima Shakeel
- Ibtisam Al Shibli
- Nabeela Badursha
- Ibrahim Al Zakwani

Conference Agenda

Day 1: Wednesday 18th February 2015	
SESSION 1: INTRODUCTION	
08:00 - 09:00 am	Registration
09:00 – 09:10 am	Conference Opening
09:10 – 09:20 am	SQU speech, HE Dr. Ali Saud Al Bemani, VC
09:20 – 09:30 am	ITA speech, Dr. Salim Sultan Al Ruzaiqi, CEO
09:30 – 09:40 am	FOSSC – Oman 2015 speech, Prof. Hadj Bourdoucen
09:40 - 10:00 am	Patrick Sinz: FOSS as a Motor for Entrepreneurship and Job Creation
10:00 - 10:30 am	Exhibition Opening / Coffee Break
SESSION 2: FOSS & CAREER	
10:30 -11:00 am	Jan Wildeboer: Open is Default!
11:00 - 11:30 am	Elizabeth K. Joseph: Building a Career in FOSS
11:30 - 12:00 am	Panel Discussion - 1
12:00- 12:30	Coffee Break
SESSION 3: FOSS & GOVERNMENT	
12:30- 01:00 pm	Hassan Al-Lawati: FOSS National Initiative in Oman, Facts and Challenges
01:00 - 01:30 pm	Flavia Marzano: The Role and Opportunities of FOSS in Public Administrations
01:30 – 02:00 pm	Panel Discussion - 2
02:00 pm	Close of Day 1 - Lunch

Day 2: Thursday 19th February 2015	
SESSION 4: FOSS & COMMUNITIES	
08:00 - 08:30 am	Registration
08:30 - 09:00 am	Matthias Stürmer: Digital Sustainability of Open Source Communities
09:00 - 09:30 am	David Hurley: Scaling Applications for Global Communities
09:30 - 10:00 am	Panel Discussion - 3
10:00 - 10:30 am	Coffee Break
SESSION 5: FUTURE OF FOSS	
10:30 - 11:00 am	Wolfgang Finke: National FOSS Strategy and the Supporting Role of Research Institutions
11:00 - 11:30 am	Patrick Sinz: Making Open Source work in and for the Public Sector
11:30 - 12:00 am	Abdullah Al Barwani : Government Open Source Cloud (G-Cloud Project, ITA)
12:00 -12:30 am	Panel discussion - 4
12:30 - 12:40 pm	Coffee Break

SESSION 6 : PAPER PRESENTATIONS	
Time	Paper Presentation
12:40 -12:50 pm	Governing Public-Private Tensions of Open Source Software Communities in the Higher Education Sector Wisal Al Bulushi
12:50 - 01:00 pm	Proposal for the Component-Based Integration Infrastructure in the Context of Moodle Environment as an E-Learning Open Source, Yusra Al-Roshdi and Zuhoor Al-Khanjari
01:00 - 01:10 pm	Evaluation of Open Source Software with QualiPSO OMM: a case for Bungeni and AT4AM for All Kennedy Ndenga Malanga , Jean Mehat , Ivaylo Ganchev, John Wandeto and Casper Shivachi Shikali
01:10 - 01:20 pm	Open-Source Linguistic Techniques for Knowledge Discovery Processing Khaled Abdalgader and Mohamed S. Hajji
01:20 - 01:30 pm	Augmented Reality for Tourism in Oman Using Free Open Source Software Ahmed Al Maashri, Sara Al-Asadi, Muna Tageldin, Shurooq Al-Lawati and Ali Al Shidhani
01:30 - 01:40 pm	Open Source Software for Arabic Citation Engine: Issues and Challenges Saleh Alzeheimi, Akram M. Zeki and Adamu I Abubakar
01:40 - 01:50 pm	System Quality Characteristics for Selected Mobile Platforms Ahood Al-Darmaki ,Nabeela Badursha ,Ibtisam Al Shibli and Mohamed Sarrab
01:50 – 02:00 pm	Open Source Software Support for Field Experiments of Vehicular Ad Hoc Networks Shaima M. Shakeel, Osama M. H. Rehman, Mohamed Ould-Khaoua and Hadj Bourdoucen
02:00 – 02:10 pm	A review of CRC mechanisms and FOSS algorithms Mahdi Amiri-Kordestani, Afaq Ahmed and Hadj Bourdoucen
02:10 – 02:20 pm	About a FOSS for the Teaching of FORTRAN Programming Language Course Afaq Ahmad, Tariq Jamil, Saad Ahmad and Amer Arshad Abdulghani
02:30 pm	Close of Conference & Lunch

Workshops Agenda

In parallel with the conference a number of workshops tailored with the needs of local communities have been scheduled. FOSS experts from Industry and private and government sectors will present these workshops.

Day 1: Wednesday 18th February 2015	
10:30 – 11:30 am	Title: Linux in Education, a Case Study: How to shift a school's computer course from application to concept learning Presenter: Feras Abou Shackra, Managing Director, Linux Professional Institute Middle East.
11:30 – 12:30 pm	Title: Using Free Software in Schools Presenter: Fahad Al Saidi, Project Specialist, ITA
12:30 – 01:30 pm	Title: Introduction to Raspberry PI Application Presenter: Anood Al Muharbi, Engineering Village
Day 2: Thursday 19th February 2015	
09:00 – 10:00 am	Title: FOSS Tools in Public Key Infrastructure (PKI) Project. Presenter: Mahmood Al Hinaai, PKI System Executive Engineer, ITA
10:00 – 11:00 am	Title: Full e-learning Process Using Moodle Presenter: Ghalib Al Abri, CEO, Rafed Company
11:00 – 12:00 pm	Title: Introduction to The World of Arduino Presenter: Suleiman Al Habsi, Engineering Village
12:00 – 01:00 pm	Title: Blender Tools for Architecture Design Presenter: Amna Al Munthari, Member of Free Open Source Students Society (FO3S), SQU
01:00 – 02:00 pm	Title: Building Oman Government Cloud from Open Source Components Presenter: Ilja Livenson, Architect of the ITA G-Cloud Project

Invited Speakers

Patrick Sinz

Managing Director of Ethiqa SAS,
Board Member of the AFUL

Title1:
FOSS as a Motor for Entrepreneurship and Job Creation

Abstract:

Open Source solutions enable the creation of innovative and nimble IT companies, but over and above this it enables any company to experiment, innovate and accelerate its development while reducing vendor lock-in and motivating internal teams. The talk will touch the issue of "making money with free" for IT centric companies, and how "the freedom of experimenting brought by FOSS solutions" unleashes benefits for all companies, and make it sustainable.

Title2:
Making Open Source work in and for the Public Sector

Abstract:

Open Source solutions enable the public sector to get a control on its IT and provides long term sustainable solutions, it also provides a simple way to ensure that public spending benefits the private economy in a transparent and ethical way. But these benefits are not without challenges. The talk will touch inter-agency collaboration, FOSS driven public private collaborations, FOSS neutral tenders, and sustainable IT.

Bio:

Patrick Sinz is the CTO of Apila SA, a product startup focusing on innovative Smart Cities Applications, and the Director of Incubation at Next Gen SAS focusing on innovative Open Source projects in the Public Sector, Smart City and Sustainable development. Patrick is a board member of the AFUL, Director of the Digital Freedom Foundation, and Founding member of Software y Cultura Libre in Bolivia. He also runs the Data Tuesday Barcelona Community event in Catalunya.

Patrick holds a DEA of Computer Science specialized in Artificial Intelligence from Paris 8, and his current interests cover Open Source solutions for the Smart City and Mobility sector, and Open Source collaboration and business models for sustainable development.

Jan Wildeboer

EMEA Evangelist at Red Hat

Title: Open is default!

Abstract:

Open is default! No matter if you call it Open Source, Open Innovation, Open Knowledge, Open Standards -The Open Source Way is becoming default everywhere. What the Next Big Thing, if such a thing exists at all? Jan Wildeboer, Red Hat's EMEA Evangelist will share his provocative observations from the past to today and beyond to find out how open is needed.

Bio:

«Jan Wildeboer works at Red Hat as the EMEA Evangelist, who is responsible for high-level customer relations and for strengthening the Red Hat brand and ecosystem. Wildeboer›s community involvement and continuous championing of open source has resulted in the growing awareness and adoption of open standards in the Netherlands and across Europe. A well known lobbyist for open source in Brussels, Wildeboer has led the successful movement to abolish the Software Patents directive in the European Parliament. He also initiated global construction technology manufacturer Hilti's migration of mission-critical systems to Red Hat Enterprise Linux.

Previously working as a Systems Administrator at Seijsener in the Netherlands and Back end Systems Software Engineer at Domain Factory in Germany, Wildeboer continued his career as Core Developer at osCommerce, the leading open source e-Commerce platform. He then moved to Red Hat Germany to become a Solution Architect. Wildeboer is a member of the Open Forum Europe, Open Source Business Foundation (OSBF) and European Committee on Interoperable Systems (ECIS).

Elizabeth K. Joseph

Linux Systems Administrator, HP OpenStack Infrastructure team

Title: Building a Career in FOSS

Abstract:

As a full-time FOSS contributor, Elizabeth has built a career around contributing to FOSS. In this talk she discusses this career, including social and volunteer involvement in projects and user groups and early jobs she had that focused on using FOSS.

Bio:

Elizabeth K. Joseph is a Linux Systems Administrator with HP working on the OpenStack Infrastructure team. She is also a long time contributor to Debian and Ubuntu, and currently holds a position on the Ubuntu Community Council and is the co-author of The Official Ubuntu Book, 8th edition. She is a member of the Board of Directors for Partimus, non-profit putting Linux-based computers into schools in San Francisco, California.

Hassan Al-Lawati

E-Industry Development Director, Information Technology Authority

Title: National FOSS Initiative in Oman, Challenges and Facts

Abstract:

The Free and Open Source Software National Initiative is Part of the e.oman strategy, to support all applications and technologies used to develop the IT Sector in Oman. It was launched in March 2010 during the Free and Open Source Software Symposium in Muscat. The broad objective of the initiative (FOSSI) is to encourage innovation, boost software development talent & skills and create the necessary resources and infrastructure to increase commercialization and industry adoption.

Bio:

Hassan Fida Al-Lawati has 16 years' work experience with a qualification of MBA degree in Business Administration from University of Bedfordshire of UK, and a bachelor degree in Marketing from SQU. He is currently the e-Industry Development Director at ITA and manages several projects that contribute to connecting the Omani manpower force and capabilities to ICT industry.

Flavia Marzano

Lecturer at La Sapienza Rome University

Title: The role and opportunities of FOSS in Public Administrations

Abstract:

Public Administrations (PAs), when acquiring software must assure: pluralism, competition, security, integration with software already in use, business continuity, interoperability, availability of source code (at least for inspection), software portability and reuse for other PAs, possibility to export data and documents in open formats.

Bio:

Falvia is a Strategic Adviser for Local Public Administration especially on Open Government, eParticipation, Cloud Computing, Social networking and FOSS adoption and regulations. She is a member of the Italian Open Source committees; head of the FOSS Competence Center of the Province of Rome; evaluator and reviewer of IST European Projects; lecturer at La Sapienza Rome University (Technologies for Public Administrations); member of the Scientific Committee of Master in Open Source Software Management (University of Pisa); and a member of the national experts committee in support to Italian Digital agenda agency. Falvia is active in the definition of Italian eGovernment and eParticipation Action Plans and related call for projects and evaluations. She holds MSc in computer science.

Matthias St¸r mer

Head of Research Center on Digital Sustainability at University of Bern,

Title: Digital sustainability of open source communities

Abstract:

What is digital sustainability and what do open source communities have to do with it? The talk introduces the concept of digital sustainability, discusses characteristics of digital resources that make them sustainable, and explains why and how communities of open source communities create digitally sustainable software. Examples of different community activities such as the LibreOffice project illustrate how collaboration works in various open source initiatives.

Bio:

Dr. Matthias Stürmer studied business administration and computer science at University of Bern until 2005 (lic.rer.pol) and finished his doctoral dissertation at the Chair of Strategic Management and Innovation at ETH Zürich in 2009 (Dr. sc. ETH Zürich). His research focused on open source communities and firm involvement, and after finishing his PhD he then worked at Liip AG, a Swiss software company creating agile Internet solutions based on open source technologies.

Among other topics Matthias Stürmer consulted global corporations and government authorities on social media governance, open source software, and open government data. He is a member of the board of Swiss Open System User Group / ch/open, member of the board of Opendata.ch, secretary of the Swiss Parliamentarian Group for Digital Sustainability, and leader of the OSB Alliance Working Group Office Interoperability.

David Hurley

Joomla Community Manager

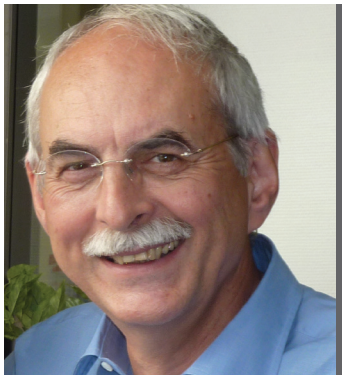
Title: Scaling Applications for Global Communities

Abstract:

For the past 10 years, the Joomla project has been a strong model and example of distributed leadership and community building in the open source landscape. In this presentation, David shares the lessons learned from their first 10 years, and provides critical insights to those working within open source projects that can assist in their success.

Bio:

David Hurley is an open source advocate. He is the founder of Mautic, the free and open source marketing automation platform, and the Community Manager for Joomla - the second largest content management system in the world. He volunteers as a member of the production leadership team, the framework maintainers and other groups. David writes obsessively at http://dbhurley.com and is an active partner in several businesses.



Wolfgang F. Finke

Business Management - Business Information Systems - eCollaboration - Educational Management

Title: National FOSS Strategy and the Supporting Role of Research Institutions

Abstract:

Free and Open Source Software, can - and will - play an important role in businesses and public organizations as part of standard IT/IS infrastructures. The reported amounts of savings, which can be achieved through utilizing FOSS by national governments and by business, are hotly debated and difficult to compare or verify. From a broader - national - perspective, there are significant advantages of utilizing FOSS

Bio:

Prof. Dr. rer. pol. habil. Wolfgang F. Finke. studies in Business Administration and Business Information systems at the University of Goettingen and the University of Paderborn/Germany. He got his Ph.D. in 1980 in the field of Business Information Systems at University of Paderborn/Germany.

Abdullah Al Barwani

Director General of Infrastructure, Information Technology Authority (ITA)

Title: Government Open Source Cloud (G-Cloud project, ITA)

Abstract:

G-Cloud is a platform based on open source technologies that enables the Omani Government and other state authorities to locate their information systems to a central state-managed private cloud and to refrain from maintaining their own IT-infrastructures.

Bio:

Abdullah has more than 13 years' experience in information technology and security management. He Obtained his Bachelors degree in computer science form the USA and focused his career in information security and obtained important certifications such as MCSE, CISM and SABSA. He worked for the Ministry of Defense for 6 years in different specializations before he was appointed the Chief Information Security Officer (CISO) as a reward for his great achievements. Abdullah established the first information security management program in the ministry and initiated an awareness and training program for the staff covering multiple regions.

He joined the Information Technology Authority (ITA) in 2006 where he started his management role as a director and then a Director General of Information Security Division supervising a national program to enhance information security within the government of Oman as part of the National eGovernment initiative.

With a career shifted to management, Abdullah completed his MBA in 2013 in UK giving him a new perspective to management that is business driven which enables him to have strong management skills even outside the technical domain. Abdullah came back to be appointed as the Director General of the Infrastructure Division; supervising major national projects such as the Government data center, the national contact center, and the Government Cloud.

Abdullah is also a committee member in many national critical committees such as the national Public Key Infrastructure Steering Committee and the National Business Continuity and Disaster Recovery Steering Committee

Feras Abou Shackra

Managing Director, Linux Professional Institute Middle East

Workshop Title: Linux in Education, a Case Study: How to shift a school's computer course from application to concept learning.

Abstract:

Imagine the possibilities of your students having access to computers without any vendor locks. Students can make whatever modification they wish, providing unparalleled flexibility. The choice is theirs, and the possibilities are limitless. It is like buying your kid a Lego toy instead of a toy car, and seeing them turning it into whatever they like. This case study shares success stories on the implementation of a Linux curriculum within grades 10 and 11.

Bio:

Feras Abou Shackra is a founding member and Managing Director of the LPI Middle East. He is responsible for driving and executing the cross-company business, policy and strategy for Linux, open standards and open source. Recent years have seen him in the movement from traditional technical and intellectual property approach, to one where business exploitation of open source, cloud and big data technologies for greater value is paramount, especially in vertical industries and emerging markets.

Feras was seen working with local and international authorities, where he has lead a UN capacity building initiative throughout 22 Arab countries. Feras is executive member of the International Standards Organization and the International forum for the IT Service Management scheme. He has led many initiatives at the LPI, which have furthered Linux and Open Source awareness – adoption and localization of internationally recognized certifications, and development of the LPI resource in a form accessible to the Arab community.

Fahad Al Saidi

Projects Specialist, FOSS National Initiative,
Information Technology Authority (ITA)

Workshop Title: Using Free Software in Schools

Abstract:

Free Software gives users the freedom to control their own computers. Free software also gives users the freedom to cooperate with each other, to lead an upright life. This can be applied and utilized in education field especially in the schools. This workshop discusses FOSS potentials in education.

Bio:

Fahad Al Saidi is a GNU/Linux supporter in Arab community. He is the founder of itwadi.com which is one of the best websites specialized in Free software in Arabic. Fahad has translated and published hundreds of articles about Open Source in Arabic. In 2012, Fahad wrote and published his first book "know Free & Open Source Software" in Arabic. Fahad is now working as Project Manager for the Free and Open Source Software National InitiativeInitiative at ITA.

Anood Al Muharbi

Instructor in the Engineering Village

Workshop Title: Introduction to Raspberry PI Application

Abstract:

The GPIO pins on a Raspberry Pi are a great way to interface physical devices like buttons and LEDs with GNU/Linux operating systems using an open source programing language (python). The workshop introduces how these applications can be used to connect projects to the real world.

Bio:

Anood is undergraduate student pursuing Bachelor's Degree in Computer Science at SQU. She is the head of the Free & Open Source Students Society at SQU, and participated in and organized several FOSS events and lectures. Anood is an active member at Engineering Village working on developing Raspberry Pi technical training program and introducing it to the Omani society.

Mahmood Al Hinaai

PKI System Executive Engineer, Information Technology Authority (ITA)

Workshop Title: FOSS Tools in Public Key Infrastructure (PKI) Project.

Abstract:

This workshop Introduces Public Key Infrastructure System (PKI) and its services provided by the National Digital Certification Center (NDCC) at ITA. It will also explains digital ENCRYPTION and SIGN, Symmetric and asymmetric ENCRYPTION methods, and differences between Public and Private Keys.

Bio:

Mahmood Al Hinaai is PKI System Executive Engineer at ITA – Oman. He is a Red Hat Certified System Administrator (RHCSA). He holds Bachelor›s degree in Electrical and Computer Engineering – Computer Systems and Networks, from SQU.

Eng. Ghalib Ali Al Abri

CEO, Rafed Company

Workshop Title: Full E-learning Process Using Moodle

Abstract:

Moodle is a learning platform that enhances the existing learning environment. As an E-learning tool, Moodle has a wide range of standard and innovative features, such as the calendar and Grade book. It is a leading Virtual learning environment and can be used in many types of environments such as education, training and development and business settings. This workshop demonstrates how Moodle can be used to design a full training process.

Bio:

Eng. Ghalib Ali Al Abri is the CEO of Rafed Group for more than 5 years. Rafed company focuses on building and designing websites and online systems using Open Source CMS software like JoomlaTM and WordpressTM. It also provides e-learning solutions and support using MoodleTM in addition to EDMS system such as OpenKMTM and AlsfrescoTM. Ghalib has a MCCC from MoodleTM. He is very interested with Arabizing software applications. He holds a Bachelor Degree in Electrical and Computer Engineering from SQU.

Suleiman Al Habsi

Instructor at Engineering Village

Workshop Title: Introduction to the World of Arduino

Abstract:

Arduino is an open-source computer hardware and software company, project and user community that designs and manufactures kits for building digital devices and interactive objects that can sense and control the physical world. The project is based on a family of microcontroller board designs manufactured primarily by Smart Projects in Italy using various -8bit Atmel AVR microcontrollers or -32 bit Atmel ARM processors. The workshop introduces Arduino board and its application as an open source project development tool.

Bio:

Suleiman Al Habsi is undergraduate student pursuing a Bachelor Degree in Computer Systems & Network Engineering at SQU. He is a peer tutor at the College of Engineering at SQU. Suleiman conducted several technical courses in applied electronics, PCB design, VB.net and Proteus VSM. Suleiman has led the R&D department at Engineering Village for 2 years and the IT student society at SQU in 2010; and was the project supervisor of IEEE-SQU in 2011. He worked with the Microsystems Design Lab, at The Pennsylvania State University (Penn State) USA for 12 weeks.

Amna Al Munthari

Member of Free Open Source Students Society (FO3S), Sultan Qaboos University

Workshop Title: Blender Tools for Architecture Design

Abstract:

The workshop explains an architectural model simulated for one of the famous institutions in the Sultanate. It provides an opportunity for the attendees to discover the most important tools harnessed to complete that project along with a practical training on how to use them. Architectural simulation in Blender focuses more on aesthetic side rather than scientific aspects.

Bio:

The Free & Open Source Software Society (FO3S) was established in 2010 under the Communication and Information Research Center at SQU and as an outcome of the MOU signed between ITA and SQU to support FOSS.

Ilja Livenson

Architect of the ITA G-Cloud Project

Workshop Title: Building Oman Government Cloud from Open-Source Components.

Abstract:

This workshop discusses G-Cloud architecture including hardware and networking aspects and concentrating on the OpenStack integration aspects and usage of open-source configuration management tools for the operations.

Bio:

Ilja Livenson is the architect of the ITA G-Cloud project. He holds MSc in computer science from the University of Tartu, Estonia, and has 10 years of professional experience with industry and academic projects. He worked on projects on data management issues at KTH PDC, developed medical health system in Estonia, helped to extend EGEE to the Baltics, and worked on the peer-to-peer virtual-world platform.

Papers List

Governing Public-Private Tensions of Open Source Software Communities in the Higher Education Sector

Wisal Al Bulushi[1]

***Abstract*—Open source software (OSS) governance is essential for creating and sustaining a community-driven software. Unlike the current relevant research that focused mainly on OSS communities serving the horizontal domains, this paper aims to focus on the importance of redirecting the attention of the relevant literature towards the governance mechanisms practiced on the OSS communities serving the vertical domains; higher education applications in specific. Such applications are targeting specific users and designed by specialized developers as their requirements are more complex and require dedicated knowledge. This paper will emphasis on the role of governance in resolving public-private tensions, which are the conflicts raised between the contradictory agendas of the community members. The paper represents the preliminary findings of studying Kuali, an OSS community developed for higher education purposes. Addressing the public-private tensions in Kuali and understanding its governance practices will assist the higher education sector in sustaining community-driven software products.**

***Index Terms*— Governance, Higher Education, Open source Software Community.**

I. INTRODUCTION

THE open source software (OSS) domain is a multidisciplinary area that attracted the interests of academics and practitioners in various fields such as management, economics, innovation and software engineering and provides interesting avenues for exploration. As OSS communities generated new organizational forms [1, 2], one of the interesting avenues of research is OSS governance.

Research to date on governing public-private tensions in OSS communities focused on resolving issues related to building the community, collective action dilemmas and the conflicting interests of individuals and organizations. However, the main emphasis was on horizontal OSS products which serve a wider range of users. The objective of this paper is to focus on governing OSS communities serving the vertical domain; the higher education context in specific. Horizontal applications refer to infrastructural applications– such as: operating systems, web servers, email servers, and databases- where their requirements and design are standard to software developers. On the other hand, vertical applications are sector oriented, such as higher education and health care systems. Vertical applications consist of technical and functional requirements that are not universally understood [3] and attract different contributors and users. Accordingly, they experience different coordination mechanisms and establish different relationships with profit and non-profit organizations.

This research aims to develop OSS governance theory by conducting an empirical study on Kuali, an OSS community serving the higher education sector. Unlike the existing studies, this research will provide empirical evidence of OSS governance mechanisms adopted by a sponsored OSS project where the product serves a vertical domain, the contributors have a common basis of authority and the project adheres to a defined hierarchy.

II. OPEN SOURCE SOFTWARE COMMUNITY

OSS community[1] refers to individuals and organizations collaborating in order to produce a software product under OSS license. The software is developed to satisfy the requirements of individuals as well as organizations. It can be freely used, modified and redistributed [4, 5].

OSS community contributors are mainly geographically distributed and collaborate through the internet and utilize digital networks- such as emails, electronic forums, bug tracking, version control systems, etc. - in order to facilitate the collaboration. The role of the contributors covers the technical, administrative, functional and financial aspects of the community [6]. Contributors may also include passive users who are benefiting out of the software without contributing back, i.e. free-riders, as non-excludability is one of the main features of the OSS product [4, 7].

Although OSS community does not gain direct monetary benefits out of the product [8], it allows private interests to collaborate in order to emerge complementary services over the OSS product [2, 8, 9]; such as supporting software or hardware, training, packaging, maintenance and consultancy services.

The relation between OSS community and the commercial vendors is regulated through OSS license, which is considered as the most essential feature distinguishing OSS from other software products [2]. It ensures that the software is freely used, modified and shared. According to Open Source Initiative [10], the free availability of an OSS product does not mean that it cannot

[1]Doctoral Researcher, Nottingham University Business School, Nottingham, NG8 1BB, United Kingdom (e-mail: lixwaal@nottingham.ac.uk)

[1] It is worth mentioning that OSS communities are also called OSS projects, thus community and project will be used interchangeably throughout this article.

be used for commercial purposes. OSS license allows commercializing the software without appropriating it. In other words, the license disallows placing any sort of restrictions that exclude an individual or entity from benefiting from the product.

III. THE GOVERNANCE OF OPEN SOURCE SOFTWARE COMMUNITY

OSS communities emerged as social movements [2, 8, 11] initiated by individuals who were counter to proprietary software firms [11]. Today, firms are essential users and contributors in the OSS community and they consider these communities as credible partners for collaboration [8]. In addition, large information technology industries are considering OSS when formulating and implementing their market strategies [2]. For example, IBM and Sun Microsystems are recognized as active firms in supporting and releasing OSS projects [7].

The dispersed community, the diversity of sponsors and stakeholders, the conflicting interests of contributors, the uncertainty of the scale and outcome of the products, and the openness nature of the OSS created challenges on selecting the suitable governance mechanism for the OSS community.

The relevant research did not provide a precise definition of OSS governance [12], as it was defined in terms of processes, structures and values. However, scholars agreed that governance is essential for creating and sustaining a community driven software. The main purposes of OSS governance are to: 1) resolve the coordination issues between the members in general, and the contradictory agendas of the contributors in specific, 2) resolve decision making and leadership issues, 3) attract and retain contributors, and 4) decide on the future direction of the project and how the community can be sustained [12-16].

The main focus of this paper is on governing the contradictory agendas of the contributors, which is referred to in this article as public-private tensions. *Public* refers to the interests of the OSS product and community, while *private* refers to the interests of the individuals and organizations participating to the community. The aim of this research is to clarify the role of governance in resolving the conflicts resulting from the process of fulfilling different goals and integrating them to achieve the common objective of the community.

IV. LITERATURE REVIEW

As one on the main objectives of this paper is to attract the attention of scholars towards governing vertical OSS communities, it is worth presenting the current studies relevant to this research area.

A. Governance in OSS: a literature review

The OSS governance literature categorized OSS projects into autonomous and sponsored projects [13, 17, 18]. Autonomous refers to the projects founded by individuals, independent from any organizations and are self-managed, i.e. do not rely on a defined hierarchy or authority [5, 18]. On the other hand, sponsored OSS projects are the projects under the control of a profit or non-profit organization [18].

The early discussions of OSS governance focused on the collaboration mechanisms adopted by autonomous communities to manage themselves around building the OSS product [11]. Scholars argued that the OSS governance of an autonomous community is "a result of an emergent process of learning" [19:259]. It does not reflect a defined set of rules that are imposed to the community [20]; instead the community collectively agrees on the governance mechanisms in order to enhance its efficiency.

As the OSS projects developed and attracted various types of contributors, the OSS governance literature was extended to include studies contrasting the different governance modes adopted by the autonomous and sponsored OSS projects [20]. Researchers mainly explored the different governance structures and concepts adopted by the OSS communities to manage the product development process at one hand, and maintain a successful relationship with profit and non-profit organizations on the other hand [8, 14].

1) OSS governance structure

An OSS governance structure is defined as a framework that regulates the interactions between contributors [2], affects the type of participation [21] and determines the quality and success of the product [22]. An OSS community does not represent a strict hierarchy nor a completely flat structure [23]. It represents a special governance form [2]. Unlike the market structure that is based on contracts, patent and copyrights, OSS community relies on revealing the source code of the product. OSS governance structure was described in the literature as a peer-production community, network, and bazaar.

OSS communities were contrasted to commercial software firms by considering them as peer-production communities as the tasks are assigned and distributed on a self-selection and decentralized manners [14], not relying on hierarchal structures. This form of governance is more attached to autonomous OSS projects initiated and governed by individual efforts and the project development process is not restricted to timelines [24].

On the other hand, it has been discussed in the literature that despite the lack of central authority in autonomous communities, some kind of hierarchy was evident in the development of the OSS code [5]. For example, the tasks related to granting file access, fixing bugs and releasing codes, were performed in a uniformed manner. Therefore, OSS communities were described as network governance forms where the coordination mechanisms are handled by informal mechanisms and do not adhere to strict bureaucratic structures.

In contrary, Demil and Lecocq [2] argued that OSS projects do not resemble a network form. They have suggested that the nature of OSS license promoted a bazaar-like environment. The major difference between bazaar and network is that, unlike network, the identity of the agent and its previous actions are not considered as important factors for coordination. In addition, persistence is an essential attribute of network participants [25] to avoid free-riding and opportunism actions, which is not considered in the bazaar structure.

In general, OSS governance structure is a controversial topic due to the openness and non-excludability features of OSS communities. Some scholars argued that each OSS community is unique, and accordingly represents a governance structure that fulfils its requirements [23]. Others argued that OSS communities rely on a certain governance structure to start-up and manage the code

building process; however in order to sustain they adopt hybrid governance structure [2, 13, 14] to manage the complementary services as well as the core code development.

2) OSS governance concepts

OSS community started as a social movement and known to be self-managed by normative control and shared values [1, 6, 8]. Therefore, social values are considered as a precondition for any formal governance mechanism either to govern individuals within a community or organizations collaborating in productizing the OSS product [9].

The key social values that govern OSS communities are trust [26] and reputation [1, 16]. OSS community is mainly a virtual organization. Therefore, to ensure an efficient collaboration, it is essential that contributors, especially developers, trust the leadership's objectives and believe that the objectives are aligned with the collective benefits of the community.

Reputation in OSS aspect, is working towards enhancing the quality of the OSS product, and thus enhancing the reputation of the participants and OSS product, ensures the continuity of the collective collaboration. Therefore, reputation is considered as a valuable asset [27] that assists in creating an OSS community, motivating participation, and ensuring sustainability.

Ownership is another OSS concept. It is an important part of the OSS culture as it regulates who can modify the software, when it can be modified and who has the right to redistribute the modified versions. This regulation is maintained through OSS licenses and sponsorship.

Authority and control are the concepts related to leadership and decision making, which are necessary to ensure the effective participation from contributors. They are distributed into two different levels: organizational level and technical level. The organizational level focuses on coordinating between the participating actors, managing the project's future direction and resolving conflicts among participants. The technical level focuses on managing the code of the OSS, i.e. dealing with code modification and new releases.

3) OSS public-private tensions

The following is a review of the main public-private tensions discussed in the OSS literature and the corresponding governance solutions suggested by scholars.

First, the tension between OSS culture and practice is one of the earliest tensions identified in OSS communities [8]. It represents the tension between OSS ideology, or the hackers' ethos as described by Raymond [28], and the practices performed by the developers in the community. In other words, it is the tension between what the OSS developers believe in on one hand, and their actual behavior on the other hand.

Although OSS culture is based on freely revealing the source code of the software for use, modification and redistribution, developers contributing to the OSS community have various motives, especially with the increase interest in OSS development within the corporate world. One of the main practices that creates this tension is *forking*, which means reusing the source code, developing a different version and changing the direction of the project [28]. Forking does not violate OSS license, however it weakens the community especially in its early stages.

According to Raymond [28], this tension can be resolved by formal and informal procedures. The formal procedures highlight the importance of creating a balance between ownership and control in OSS projects to differentiate between official and non-official releases of the OSS project. The informal procedure is exemplified in leveraging implicit rules of OSS culture that illustrates the importance of reputation in an OSS community.

Second, the tensions between the collective interests of the OSS community on one hand and the private interests of individuals or organizations on the other hand is one of the topics that has been discussed extensively in the OSS literature. However, the main focus was on the autonomous OSS projects with voluntarily participations.

One of the solutions suggested by scholars for this tension is an adaptive governance scheme that changes as the OSS community grows [14]. OSS communities often start with a meritocracy governance mode, i.e. a centralized leadership, due to the low number of contributions at that early stage. Then, as the OSS project increases in size and attract contributors with various interests, it tends to implement different governance modes to fulfil the new requirements of the community, and accordingly balance between satisfying different interests.

Third, the tensions emerge between developers of different motives. This is referred to by Franck and Jungwirth [27] as the tension between *donators* and *rent-seekers*. *Donators* are voluntarily developers, or hobbyists, who are contributing for the benefit of the group and not expecting to receive any rewards out of their contributions, while *rent-seekers* are those who act in a self-interest manner aiming to invest from their contributions.

It is essential to reconcile the interests of donators and rent-seekers as the OSS community relies on both of them [27]. This can be done by incentivizing developers with self-interest motives in order to collaborate in more efficient manner [22]. Another solution is achieved by adding features to the OSS license which disallows contributions from turning donations into private profits without contributing to the production of the OSS [27]. As a result, rent-seeking is enabled without crowding out donators.

Fourth, OSS communities face challenges in resolving the tension between openness and ownership. OSS openness refers to the transparency and accessibility features [18] of the community. On the other hand, OSS ownership deals with the regulations and restrictions and it is exemplified by the OSS licensing and organizational sponsorship.

OSS openness-ownership tensions were discussed in the literature from developers' attraction [18, 29] and coordination [2] perspectives. The process of attracting developers while balancing between OSS openness and ownership is a challenging matter for both autonomous and sponsored projects. The suggested governance solutions rely on the community design decisions. Autonomous communities tend to rely on licenses and sponsorship to create the balance between openness and control [29], whereas sponsored communities rely on the transparency and accessibility to the code [18] to attract developers.

From coordination perspective, Demil and Lecocq [2] argued that OSS communities require a governance structure to regulate the transactions between different actors. They agreed with Raymond [11] that OSS communities represent a bazaar governance structure which is compatible with the OSS license features. Although bazaar governance is

suitable for information goods; pure bazaar is a failure due to the existence of uncertainty and weak control. Therefore, hybrid governance, which combines bazaar with hierarchy or network structures, may resolve the coordination aspect of openness-ownership tension.

Fifth, tensions between the OSS community and the commercial vendors are a result of their divergent objectives. In addition, the core resources of an OSS project reside within the community, whereas the marketing, sales and distribution capabilities reside within the firm [30].

Scholars argued that such tension is resolved by forming foundations [8], creating specific business models for OSS products [3, 7] and facilitating the value creation and capturing mechanisms in OSS projects [31].

4) *Kuali: a literature review*

Kuali project was initiated in 2004 [32] and gained the interests of scholars in areas of technology adoptions in HE [33], social sciences [34-37], finance [38] and software engineering [39].

From social sciences perspective, current research explained the organizational control mechanisms practiced in Kuali using Ouch's (1979, 1980) organizational control framework [as cited in 36]; however the findings were not supported by empirical evidence. In addition, scholars explained the factors that influenced the process of building Kuali community in its early stages [37] and discussed the challenges of in-house staffing [35].

The current research on the social sciences aspect of Kuali is either not supported by data or reflects the early stages of forming Kuali community. It is evident that the literature overlooked the public-private tensions occurring within the community and those between the community and firms.

5) *Summary of the literature review*

The literature on OSS governance mainly focused on autonomous projects and how members get to share a basis of authority and govern themselves, overlooking the governance mechanisms adopted by sponsored OSS projects which already have a basis of authority and hierarchy.

The focus of governing public-private tensions in the literature has moved away from tensions raised within the community while developing the OSS code towards the tensions beyond the boundaries of the community to govern complementary services of sponsored OSS projects. However, the main focus was on communities built around horizontal applications, such as Debian [e.g. 1, 8], Linux [e.g. 1, 11] and Apache [e.g. 1, 8].

The existing research on vertical domains compared between autonomous and sponsored OSS projects in the health care systems in terms of building the community [17]. In addition, the main focus was on the tensions raised on the developers' level with less attention given to the macro-level.

V. RESEARCH OBJECTIVES

This paper aims to explain the role of governance in resolving public-private tensions in OSS communities serving the higher education sector. The main objective is to redirect the attention of the OSS literature towards vertical domains, as the primary focus of the relevant studies was on horizontal domains.

Unlike horizontal OSS communities, vertical communities serve particular domain where participants are obliged, and paid in some cases, in order to contribute to the community as opposed to the voluntarily contribution of the participants in the horizontal domain. In addition, horizontal domains attract the interest of generic participant regardless of their profession. In contrary, vertical domain are sector oriented and include participants involved on that particular sector. Vertical domains are experiencing different coordination, motivation and collaboration mechanisms that are worth exploring.

A. *Why higher education?*

The research targets the higher education (HE) sector for two main reasons. First, the values of OSS community align with the nature of the HE institutes. This is exemplified in collaboration, knowledge sharing and capacity building.

Second, "Higher education is a small segment of the overall technology market, and the needs of the academy may not be primary to a company" [40:10]. As a result, the HE often falls into the *buy-build* dilemma when adopting an IT solution. It struggles between the cost and benefits of building their own solution or going for a commercial package [40]. Some universities went for a third option; collaboration or borrow [41]. This is exemplified in OSS solutions. The main three motives for HE sector to adopt OSS solutions are: cost, performance and control [42].

It terms of cost, OSS is not free of cost; however the cost in this case will be invested in the HE resources. With regards to performance, OSS communities require HE institutes to pool their resources, experiences and capabilities to improve their services and share the risks. Control is what is described by Wheeler [41] as *unbundling of software and support.* In other words, OSS solution unbundles the software development from the maintenance and support processes. Cost, performance and control are achieved by appropriate governance mechanisms.

By studying Kuali, the research aims to understand the governance mechanisms adopted by Kuali to build the community, attract divergent contributors and ensure sustainability.

VI. METHODOLOGY

The research and theory regarding public-private tensions in vertical OSS communities are in their early stages. Therefore, a grounded theory approach is adopted using a case study method. A grounded theory approach, as described by Charmaz [43], assists in developing theories from data and accordingly this study involved the collection of rich data to provide a clear description of people, organizations and processes related to the case study.

The case study, Kuali, consists of universities and firms that are geographically dispersed and collaborate through the internet and utilize technological tools to facilitate their collaboration. Therefore, Kuali website is considered as a rich source of secondary data, such as: mailing threads, videos, technical and functional documentations and discussion forums.

This article represents the initial findings of the study based on analyzing firms' portfolios, mailing threads, and technical and functional documentations of Kuali foundation.

VII. PRELIMINARY FINDINGS

Kuali Project is an OSS project governed by the non-profit Kuali Foundation to develop applications for higher education, by higher education. Kuali has produced several OSS projects, mainly: financial system, research system, student information system, payroll system, library system, smart phone framework, and a middleware application to integrate all Kuali applications together.

According to Kuali Foundation [32], the community consists of HE institutes (i.e. universities and research centers), non-profit organizations and commercial vendors that collaborate to produce OSS products as well as complementary services to support the community. Contributors are categorized into different levels, such as members, partners and adopters. Each has certain privilege to benefit from the community besides having the OSS products of Kuali freely available for all levels to use, modify and distribute.

Although there exist OSS projects in higher education communities other than Kuali, (e.g. Sakai, UPortal, Moodle), Kuali has been selected for its features that are considered as rich data for research. Kuali consists of multiple standalone projects where each has its community, life cycle, project hierarchy, mailing lists, documentation and complementary services. The contributing universities have the option to freely download Kuali OSS products from Kuali website; however 74 universities and 11 commercial firms to date [32] are paying annual membership fees to contribute and benefit from Kuali community.

Kuali developers are employees working for the contributing universities and firms. Developing Kuali products is a task added to their job responsibilities. Contributing to the code development is performed within written and signed agreements. Each Kuali product has a project hierarchy and a defined work flow.

By analyzing the documentation, it is evident that there exist public-private tensions in the micro and macro levels of the community. For example, the openness-ownership tension is evident in revealing the source code of the products on one hand and setting specific licensing terns to protect Kuali rights on the other hand. The tension between donators and rent-seekers is also evident in email conversations between developers working for a university (donator) and developers working for firms that are providing software solutions (rent-seekers).

This indicates that Kuali Foundation is adopting governance mechanisms which harness the efforts of the community and the firms to succeed and sustain.

VIII. CONCLUSION & FUTURE WORK

This research has theoretical and practical implications. In terms of theory, the empirical findings will establish a new research direction in the field of OSS governance where the target community serves a sector oriented domain and inherits the governance structures and concepts of that particular sector. In practice, OSS communities are considered as a solution to the buy-build dilemma in the HE sector. It is essential to ensure the sustainability of such communities. Understanding the governance mechanisms to resolve public-private tensions will secure the collective interest of the community. It will also attract commercial complementary contributions to support the community.

The work presented in this article is part of an ongoing PhD research. Therefore the future work includes representing a detailed description of the public-private tensions experienced by Kuali community and what are the governance practices adopted by Kuali foundation to manage these tensions. Future work will also include the analysis of videos illustrating important events of Kuali community. In addition, future work will also involve a detailed description of the coding process of the grounded theory approach.

REFERENCES

[1] Markus, M.L., B. Manville, and E.C. Agres, *What Makes a Virtual Organization Work?* Sloan Management Review, 2000. **42**(1): p. 13-26.

[2] Demil, B. and X. Lecocq, *Neither Market nor Hierarchy nor Network: The Emergence of Bazaar Governance.* Organization Studies, 2006. **27**(10): p. 1447-1466.

[3] Fitzgerald, B., *The Transformation of Open Source Software.* Mis Quarterly, 2006. **30**(3): p. 587-598.

[4] O'Mahony, S., *Guarding the Commons: How Community Managed Software Projects Protect their Work.* Research Policy, 2003. **32**(7): p. 1179-1198.

[5] de Laat, P.B., *Governance of Open Source Software: State of the Art.* Journal of Management & Governance, 2007. **11**(2): p. 165-177.

[6] Nelson, M.L., R. Sen, and C. Subramaniam, *Understanding Open Source Software: A Research Classification Framework.* Communications of the Association for Information Systems, 2006. **17**(12): p. 266–287.

[7] Dahlander, L., *Appropriation and Appropriability in Open Source Software.* International Journal of Innovation Management, 2005. **9**(3): p. 259-285.

[8] O'Mahony, S., *Nonprofit Foundations and their Role in Community-Firm Software Collaboration*, in *Perspectives On Free and Open Source Software*, J. Feller, et al., Editors. 2005, The MIT Press: Cambridge.

[9] Feller, J., et al., *From Peer Production to Productization: A Study of Socially Enabled Business Exchanges in Open Source Service Networks.* Information Systems Research, 2008. **19**(4): p. 475-493.

[10] Open Source Initiative. *Open Source Initiative*. 2014; Available from: http://opensource.org/.

[11] Raymond, E.S., *The Cathedral and the Bazaar.* Knowledge, Technology & Policy, 1999. **12**(3): p. 23-49.

[12] Markus, M.L., *The Governance Of Free/Open Source Software Projects: Monolithic, Multidimensional, Or Configurational?* Journal of Management & Governance, 2007. **11**(2): p. 151-163.

[13] O'Mahony, S., *The Governance of Open Source Initiatives: What Does it Mean to be Community Managed?* Journal of Management & Governance, 2007. **11**(2): p. 139-150.

[14] O'Mahony, S. and F. Ferraro, *The Emergence of Governance in an Open Source Community.*

Academy of Management Journal, 2007. **50**(5): p. 1079-1106.

[15] De Noni, I., A. Ganzaroli, and L. Orsi, *The Governance of Open Source Software Communities: An Exploratory Analysis.* Journal of Business Systems, Governance & Ethics, 2011. **6**(1): p. 1-18.

[16] Di Tullio, D. and D.S. Staples, *The Governance and Control of Open Source Software Projects.* Journal of Management Information Systems, 2013. **30**(3): p. 49-80.

[17] West, J. and S. O'Mahony. *Contrasting Community Building in Sponsored and Community Founded Open Source Projects*. in *Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05)*. 2005. Hawaii: IEEE.

[18] West, J. and S. O'Mahony, *The Role of Participation Architecture in Growing Sponsored Open Source Communities.* Industry and Innovation, 2008. **15**(2): p. 145-168.

[19] De Noni, I., A. Ganzaroli, and L. Orsi, *The Evolution of OSS Governance: A Dimensional Comparative Analysis.* Scandinavian Journal of Management, 2013. **29**(3): p. 247-263.

[20] O'Mahony, S. and J. West. *What Makes a Project Open Source? Migrating from Organic to Synthetic Communities*. in *Academy of Management Annual Meeting*. 2005. Honolulu.

[21] Shah, S.K., *Motivation, Governance, and the Viability of Hybrid Forms in Open Source Software Development.* Management Science, 2006. **52**(7): p. 1000-1014.

[22] Singh, P.V. and Y. Tan, *Developer Heterogeneity and Formation of Communication Networks in Open Source Software Projects.* Journal of Management Information Systems, 2010. **27**(3): p. 179-210.

[23] Martinez-Torres, M.R. and M.C. Diaz-Fernandez, *Current Issues and Research Trends on Open-Source Software Communities.* Technology Analysis & Strategic Management, 2014. **26**(1): p. 55-68.

[24] Moore, P. and P.A. Taylor, *Exploitation of the Self in Community-Based Software Production: Workers' Freedoms or Firm Foundations?* Capital & Class, 2009. **33**(1): p. 99-119.

[25] Jones, C., W.S. Hesterly, and S.P. Borgatti, *A General Theory of Network Governance: Exchange Conditions and Social Mechanisms.* Academy of Management Review, 1997. **22**(4): p. 911-945.

[26] Gallivan, M.J., *Striking a Balance between Trust and Control in a Virtual Organization: A Content Analysis of Open Source Software Case Studies.* Information Systems Journal, 2001. **11**(4): p. 277-304.

[27] Franck, E. and C. Jungwirth, *Reconciling Rent-Seekers and Donators–The Governance Structure of Open Source.* Journal of Management and Governance, 2003. **7**(4): p. 401-421.

[28] Raymond, E.S., *Homesteading the Noosphere.* First Monday, 1998. **3**(10).

[29] Stewart, K.J., A.P. Ammeter, and L.M. Maruping, *Impacts of License Choice and Organizational Sponsorship on User Interest and Development Activity in Open Source Software Projects.* Information Systems Research, 2006. **17**(2): p. 126-144.

[30] Dahlander, L. and M. Magnusson, *Relationships between Open Source Software Companies and Communities: Observations from Nordic Firms.* Research Policy, 2005. **34**(4): p. 481-493.

[31] Morgan, L., J. Feller, and P. Finnegan, *Exploring Value Networks: Theorising the Creation and Capture of Value with Open Source Software.* European Journal of Information Systems, 2013. **22**(5): p. 569-588.

[32] Kuali Foundation. *Kuali Foundation*. 2014 08/05/2014]; Available from: https://www.kuali.org/.

[33] van Rooij, S.W., *Adopting Open-Source Software Applications in US Higher Education: A Cross-Disciplinary Review of the Literature.* Review of Educational Research, 2009. **79**(2): p. 682-701.

[34] Liu, M. and J.L. Zhao. *On Outsourcing and Offshoring in Community Source*. in *Advanced Management of Information for Globalized Enterprises*. 2008. IEEE.

[35] Liu, M., et al., *Outsourcing of Community Source: Identifying Motivations and Benefits.* Journal of Global Information Management, 2010. **18**(4): p. 36-52.

[36] Liu, M., S. Hansen, and Q. Tu. *Organizational Control in Community Source Projects*. in *SIGBPS Workshop on Business Processes and Services (BPS'12)*. 2012.

[37] Liu, M., C. Hull, and Y. Hung. *Antecedents of Community Source Network Formation: The Case of Kuali*. in *46th Hawaii International Conference on System Sciences (HICSS)*. 2013. Hawaii: IEEE.

[38] Liu, M. and J.L. Zhao. *Real Options Analysis of The Community Source Approach: Why Should Institutions Pay for Open Source*. in *Proceedings of the First China Summer Workshop on Information Management*. 2007.

[39] Liu, M.L., H.J. Wang, and J.L. Zhao, *Technology Flexibility as Enabler of Robust Application Development in Community Source: The Case of Kuali and Sakai.* The Journal of Systems and Software, 2012. **85**(12): p. 2921–2928.

[40] Brooks, L. *Considering Open Source: A Framework for Evaluating Software in the New Economy*. EDUCAUSE Center for Analysis and Research (ECAR), 2007.

[41] Wheeler, B. *The Inevitable Unbundling of Software and Support*. 2004.

[42] Dolphin, I. *Open Source in Higher Education: Building a Life Raft for the Perfect Storm*. 2014.

[43] Charmaz, K., *Constructing Grounded Theory*. 2nd ed. 2014: Sage.

Proposal for the Component-Based Integration Infrastructure in the Context of Moodle Environment as an E-Learning Open Source

Yusra Al-Roshdi[2] and Zuhoor Al-Khanjari[1]

Abstract— **E-Learning is the technology that is used for education purpose. The new and increased demands of education support the needs for more efforts to develop and improve this technology. This research focuses on some limitations regarding the existing software tools for practical sessions of Computer Science Education (CSE) in the Learning Management Systems (LMS). This problem is affecting the education in general and CSE as a special case. Software tools integration infrastructure is proposed to solve this limitation in the Open Source LMS: Moodle by using the component-based as the main integration approach that benefit from Moodle features as open source.**

Index Terms— **Component-Based, Integration, Open Source, Moodle, Software tools**

IX. INTRODUCTION

E-learning is the new revolution in Education area. It supports the traditional learning and be a complement of it. Therefore, it needs a special effort to prepare its applications to fit the education purpose. Computer Science Education (CSE) is one area that uses the aspects of e-Learning to educate learners and to distribute knowledge to them. It involves theoretical and practical part for teaching its courses. Many LMSs are supported the theoretical content of the course. However, the software tools that support the practical sessions are not applied in most of the LMS. Therefore, CSE needs more effort to design its e-Learning application by integrating it with more graphical applications tools that could encourage learning by sharing knowledge and resources in the practical part. It is worth to state that the development of the online courses for the CSE is still considered as a difficult task. Although there are large varieties of software tools available for instructors of CSE, they are not integrated nor prepared completely as virtual labs [1].

This paper focuses on the development of a feasible system architecture prototype that extends the needed tools by integrating them as a part of the designed on-line courses for the CSE. The focus will be on Moodle as an open source LMS to help in adoptions of the new components. This architectural framework is expected to be simple, scalable, distributed resources and cost effective to work in any existing e-Learning platform and mainly on Moodle as the open source environment.

[2]Department of Computer Science, College of Science, Sultan Qaboos University, P.O. Box 36, Al-Khodh 123, Muscat, Sultanate of Oman (phone: 968-24141407; fax: 968-24141407; e-mail: m095653@student.squ.edu.om and zuhoor@squ.edu.om)

In the literature there are many attempts to solve this limitation regarding the practical sessions. One of them is Component-Based Approach. This approach is developing a new component in the LMS with same programming languages of it. This solution allows the users to use the tools needed for the practical sessions along with the online material in the same LMS environment.

In this paper, the Component-Based Approach will be investigated and applied on the Moodle environment from different perspectives such as: Design and Architecture, Implementation of proposed approach and the advantages and disadvantages according to the experimentation. Section II presents some related work. Section III introduces an overview about the open source in the e-learning. Section IV shows the design architecture and development method of proposed approach. Section V provides conclusions and future work.

X. LITERATURE REVIEW

Learning is one area that influenced by the technology revolution. The e-Learning is new revolution in education. Most of researchers try to expand the functionality of online education in general and CSE as a special case. They proposed architecture integration for new software tools or use third party tools and integrate them into the LMS [2-5].

In 2003, Akingbade and colleagues discussed the need for online visualization software for CS courses [6]. In 2005, Al-Khanjari and colleagues developed architecture for extended the functionality of E-Learning portal to support CSE [7]. In the following years the researchers proposed software tools to auto correct the programming languages in the LMS environment (e.g. Moodle) [8-10]. Georgantaki and Retalis showed some software tools for teaching CS courses, and they stated that there are difficulties of studying the CS courses without using the software tools for practical sessions to support the theory concepts [11]. For instance, to manage the theory content there is e-Portfolio tools such as: Google Drives, Wiki-spaces, blogs and others, which could be integrated into the e-Learning platforms [12]. In 2009, Robling and colleagues tried to improve software tools for CS exercise visualization [13]. Therefore, one of the new trends is to use online web tools for teaching, adopt that tool and integrate it into LMS [1].

In addition, Joint Information Systems Committee (JISC) is a research organization to support the education by using the new technology [14]. In 2013, Deperlioglu and Kose introduced a blended e-Learning for teaching "Data Structures and Algorithms" which included practical and theoretical topics that principally computer programming

students must know. A combination of face-to-face and online learning environments is provided to perform educational activities [15].

The component extension of any software system was introduced in 2005 by Doberkat and colleagues. The concept "E-Learning Software Engineering" was presented in their work. It focused on providing software tools for software course. The project was called "MuSoft" [16]. It works as a portal that they integrated other tools in it.

In 2008, Corbara and colleagues described new module in Moodle to support teaching computer organization course [17]. This Module carried a wide range of activities and assessments in computer courses.

In 2009, Rößling and Vellaramkalayil proposed new Module in Moodle for algorithm course. This module depends on providing tools that help in teaching algorithm course [18].

Furthermore, Prieto-Blazquez and colleagues proposed an integrated structure for virtual laboratories of programming in Computer Engineering and Software Engineering. They discussed the issues regarding the use of that laboratory in teaching the practical work [19, 20].

In 2011, Fest proposed integration of Java applet tools to help in learning the Dynamic Geometry Software (DGS) then integrated as a component into the interactive learning activities [21].

In 2012, Lavrishcheva and colleagues proposed a website portal that supported teaching Computer Science courses using different tools [22].

In 2013, some researchers explained how to extend the functionality of Moodle to facilitate the assessment of Matlab programing [8].

XI. E-LEARNING AND OPEN SOURCE LMS

There are many LMS platforms available in two main categories of LMS: 1) Open source initiatives and 2) Commercial platforms. The importance of open source in e-Learning appears in the customization of that LMS by designing tools of the courses according to the students' needs. Therefore, the open source LMS helps in the adoption of any requirements of the education area in its platform [23].

Most organizations choose an open source e-learning platforms for many reasons:

1. The commercial e-Learning platforms need to rely on the provider of that LMS for any maintenance or customization.
2. As code is available in open source LMS many developers use that feature to test the code after any modification or integration of the new component.
3. Open source LMS allows the users to customize the platforms according to the organization brands and style.
4. It helps in integration of the new component using different integration approaches as mentioned in [24, 25].

Open source LMSs are typically built on extendable frameworks allowing developers to adjust and modify the LMS to suit their specific needs. Nowadays, many LMSs make their source code available for modifications under the various displays of open source licenses and for customization to fit the users' needs [26]. The development of the e-Learning area needs to be integrated and adopted with many resources and different kinds of subsystems to support learning process [27]. As in the traditional e-Learning application, the system is monolithic as one unit system. However, the modern system needs to be more flexible and supports customization. This could be achieved since open source system provides the source code [26].

Moodle is the most successful LMS as an open source. This feature helped in facilitating the development and maintenance of the Moodle according to the users' needs. On the other hand, it provides a research area to extend the software tools of the CSE [25, 28]. Therefore, this research paper concentrates on the Moodle as the area of development.

A. Moodle

Moodle is a Learning Management System (LMS) and stands for Modular Object-Oriented Dynamic Learning Environment. It is an open source system under General Public Licenses (GNU).

Moodle application is organized as core system with many plugins as the users' need to support different functionality such as: Assignments, Quizzes, Calendar, Themes, Registration, Course and Activity, etc. Moodle is developed in PHP programming languages. It uses the common three tier client/server architecture [28]. Add to that Moodle is based on object-oriented programming.

Each organization needs to move towards the customization of its e-Learning application according to the goals and targets of that organization. The extension of new tools is one of those customization features [31]. The features of being open source will facilitate the development and the maintenance of it according to the customers' needs. This provides a research area to extend the software tools of the CSE [28].

1) Moodle Advantages and Limitations

Moodle is chosen for the following reasons [32]:

1. It is open source software that allows the modification of the code and distributes it under General public Licenses GNU.
2. Compatible with a range of 75 languages in 175 countries.
3. It has excellent documentation and strong supporting security and administration.
4. It can be run in any system that supports PHP languages.
5. It uses a single database to store data.
6. It is available in most of learning institutes.
7. It can support unlimited users.
8. It has strong plug-ins features and supports integration of external tools.

The Limitations of Moodle include:

1. Non-IT Teacher will face difficulties to install and use Moodle or customize it.
2. The installations manuals contain a lot of technical conception that are seen difficult with some technicians.

3. Moodle will not support the learning process without designing its course and support it by different software tools.

B. System Requirements

The experiment to apply the proposed architecture depends on the Moodle v2.6. This Moodle installed using XAMPP package (1.8.3). This package includes the basic requirement for Moodle installation such as web server (Apache), the database (MySQL) and PHP programming language. The experiment is conducted in environment of Windows Server 2008 R2. After preparing the Moodle environment, Moodle was installed completely with its plugins. Then the Component-Based approach applied with the Compiler software tool which called Virtual Programming Lab (VPL) to support CSE process.

XII. METHODOLOGY

1) Component-Based Approach

The solution that proposed in this research and which is introduced in some previous work [13, 16, 22, and 30] is suggested to build the software tools in the same environment of LMS. Especially, if that LMS is as open source, so the course designer will be able to customize the E-learning system while the source code will be available as open source. Therefore, the needed software tools could be built as a whole component then integrated it in the LMS using the API integration approach. This solution will let user to use the software tools in the same environment of LMS. In contrast, this approach may reduce the performance of the LMS because a lot of students will run the same software in the same time.

a) API Integration

The Component-Based architecture is based on the API integration which allows the different components in the system to interact with each other through the Application Programming Interface (API). This integration needs to define the interface for each component and then use it in the system. The interface is the signature of the methods that component offered. As Huerts and colleagues showed that there are two different approaches of integration, one of them is API integration. This approach is used to make the functionalities of the e-Learning public by defined interface. However, the integration happens in terms of a code written in the same language in which the e-Learning platform has been built. Moreover, API is used to extend the basic functionalities provided by platform or expose the external application [33].

This approach will be used to link between the LMS (e.g. Moodle) and the new established component. Therefore, after coding, the new tool must be integrated to Moodle API core using its interface.

2) Design and Analysis

The designing of the proposed approach is showing in Figure 1. It follows some process to integrate that needed tools in the LMS. First, the LMS should be analysis and studied its structure to know the LMS style and its techniques. Then, the software tools should be prepared according to the programming languages of the LMS. After that, the new component should be integrated into the LMS system by integrating it to the API of the core system. Finally, the link to the software tools should be provided in the corresponding course.

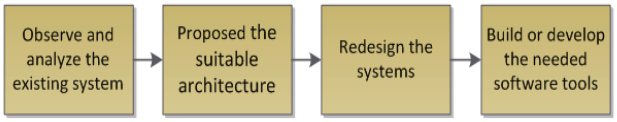


Fig 1. Development Phases of the Component-based Software Tools

Figure 2, shows the abstract overview of the proposed architecture to integrate the needed software tools using component-based approach.

There are many ways to build the software tools and integrate it in the LMS using the Component-Based Approach. The developer could search for plugins of

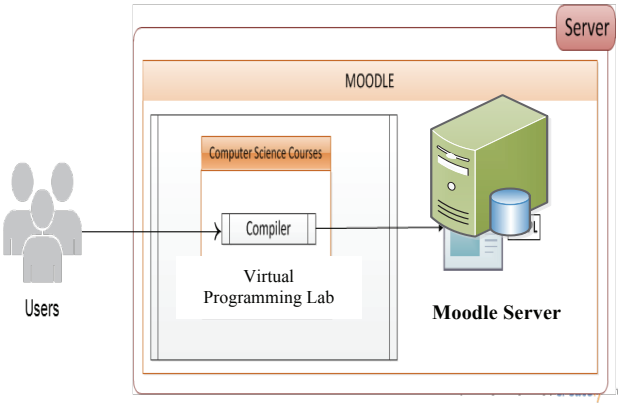


Fig 2. The abstract overview of the proposed integration approach

Moodle that provided by different providers. Then add that plugin tools to the existence plugins from the Moodle application. Moreover, the developer could develop the needed software tools and create its API that communicates with application core. Add to that, the whole block could be built to adopt more functionality to the software tools especially and course material in general.

Figure 3 displays plugin tools integration approach. This figure shows the overall architecture of the integration between the users and the component of the Moodle.

3) Experimentation on Plugin Tool

To prepare the LMS with the needed tools for the CSE such as compiler, first identify the programming language and coding style of the LMS. For instance, Moodle uses PHP as main programming language. Then, develop the needed software tools and provide it with needed functions. Next, add that tool as a new activity module and use its Application Program Interface (API) to access Moodle core so that it can access it easily from specified Moodle's course. After that, student can easily open the tool and work in the same environment.

An example of the integration using this approach is the plug-in tool called Virtual Programming Language (VPL). VPL is a compiler plug-in developed in the Moodle environment to support the CSE and any other course activities. It allows students to compile different programming languages such as: C, C++, Java, SQL, etc.

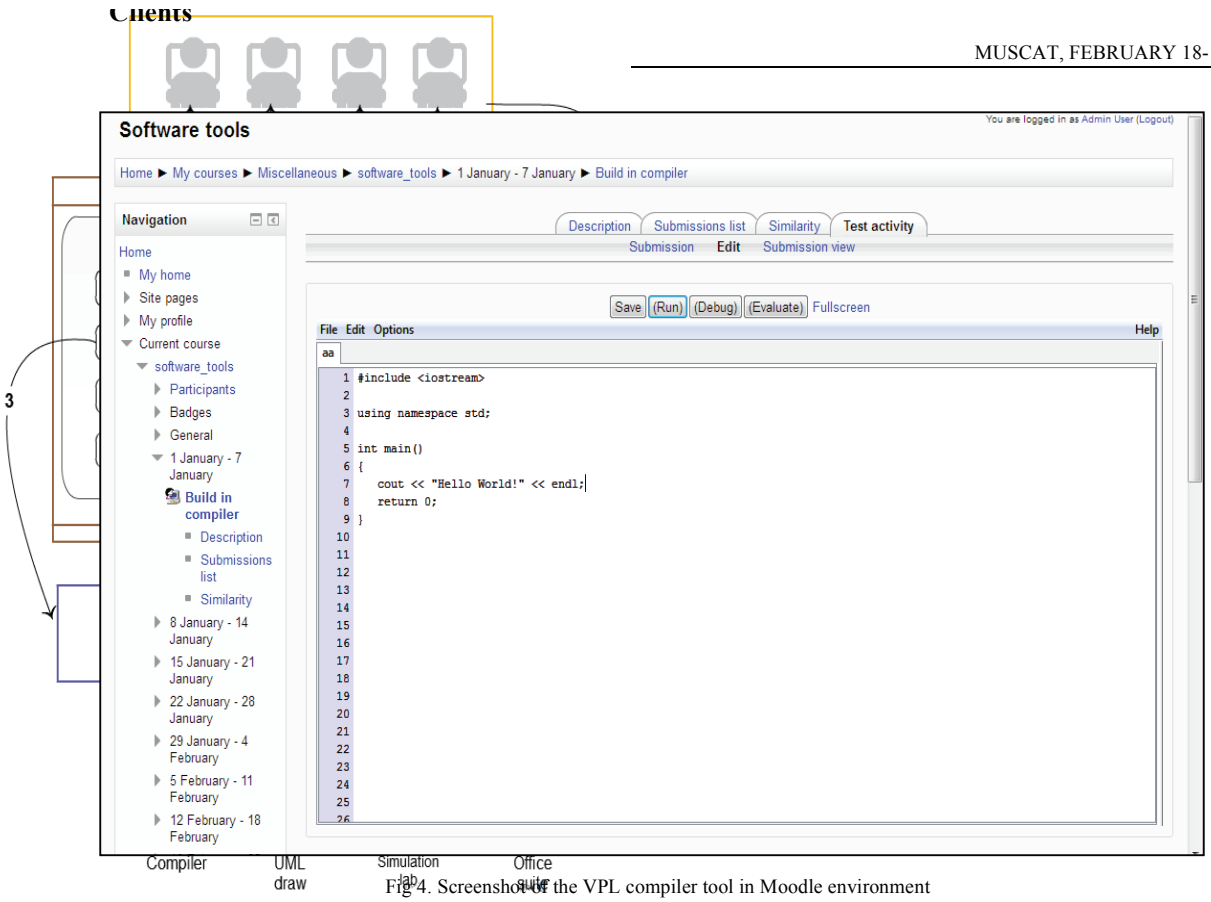


Fig 4. Screenshot of the VPL compiler tool in Moodle environment

Fig 3. The overall architecture and methodology of the proposed integration approach

Also, students can submit programming exercises and assignments through it.

This attempt may reduce the performance of the LMS and lead to a bottleneck in it since it will be utilized by the students who will run the software tools concurrently.

To develop new plugin in Moodle, the PHP language should be used in the software tool development. First, develop the tool with the its function as a component or separate project. Then communicate with Moodle core system with its APIs function. After that, the version of function should be modified to detect the new version of the system during the installation in the files *version.php*. So, the new component should be added in the file *admin_tree*. Then, the designers or developers can detect the new tools to add it to their course. In file *event.php* the event handler must be identified for new modules. Also, the database tables could be edited if the tools produce information that needs to store in the database. These changes must be updated in the files *install.xml* and *install.php*. However, the version of Moodle must be detected before and modification because it differs from one version to other. Finally the tool is available as one of the Moodle Functions, so the developers or course designer can add it easily to the related course.

Figure 4, shows the VPL compiler tool that is provided by Moodle community to the CS course students for programming practice and assignment submission.

A. Advantages and Disadvantages

Advantages:

- The proposed work in this paper helps the students to access the software tools of the CSE online from any place and at any time. This will help in improving the students' knowledge about different practical concepts of the CSE.
- It provides the needed software tools in the Moodle environment following its coding style and using PHP programming language.
- User's data can be shared between the plugins of Moodle.

Disadvantages:

- Plugging software tools could produce a bottleneck on the Moodle because the students will remain logged-in doing their work in the same environment of Moodle.
- The new added component will consume the resources of the Moodle and need a huge capacity and performance facilities.
- The new added component could reduce the performance of the Moodle, since it will be utilized by the students who are running the software tools.
- This approach could increase the development effort when they integrate the tools in the Moodle. So, it needs to redesign and recode the needed software tools.
- Teachers might not be able to develop the needed tools without the help of the developer.

XIII. Conclusion and Future Work

The study is aimed to extend the main idea behind the development of e-Learning platforms tools to be useful in the academic life. This research showed that many Learning Management Systems (LMSs) have a limitation in the

availability of the software tools for practical sessions for education in general and for computer science education as a special case. Therefore, this paper proposed architecture of the Component-Based integration approach to integrate the needed software tools into the LMS environment to help learners gain knowledge through the practical session. Also, it depends on the Moodle environment as an open source LMS to help in the integration process by accessing its code. This research could be used as the base of extending other discipline area for improving the practical session in other education area such as Engineering, Science Education, Medicine, etc. Moreover, by using the Cloud Computing (CC) technology, the material and tools of e-Learning could be shared in different LMS environments. Add to that, the security is an important issue that the system's developers should concern about in order to prepare and transfer the data in a trusted and secured way. Therefore, these issues could be studied and improved in the future.

References:

[1] Riabov, V. V. and Higgs, B. J. (2010). *Software Tools and Virtual Labs in Online Computer-Science Classes.* Learning Management System Technologies and Software Solutions for Online Teaching: Tools and Applications, 332.

[2] Baving, T., Cook, D. and Green, T. (2003). *Integrating the Educational Enterprise: Technical Report*, University of Cape Town, and Department of Computer Science.

[3] Huan, X., Shehane, R. and Ali, A. (2011). *Teaching computer science courses in distance learning*. Journal of Instructional Pedagogies, 6.

[4] Pätzold, S., Rathmayer, S. and Graf, S. (2008). *Proposal for the Design and Implementation of a Modern System Architecture and integration infrastructure in context of e-learning and exchange of relevant data.* Paper presented at the ILearning Forum.

[5] Tawfik, M., Sancristobal, E., Martin, S., Diaz, G. and Castro, M. (2012). *State-of-the-art remote laboratories for industrial electronics applications.* Paper presented at the Technologies Applied to Electronics Teaching (TAEE).

[6] Akingbade, A., Finley, T., Jackson, D., Patel, P. and Rodger, S. H. (2003). *JAWAA: easy web-based animation from CS 0 to advanced CS courses.* Paper presented at the ACM SIGCSE Bulletin.

[7] Al-Khanjari, Z., Kutti, S. and Hatam, M. (2005). *An extended e-learning system architecture: integrating software tools within the e-learning portal*. In IAJI.

[8] Ramos, J., Trenas, M. A., Gutiérrez, E. and Romero, S. (2013). *E-assessment of Matlab assignments in Moodle: Application to an introductory programming course for engineers.* Computer Applications in Engineering Education, 21(4), 728-736.

[9] Schwieren, J., Vossen, G. and Westerkamp, P. (2006). *Using software testing techniques for efficient handling of programming exercises in an e-learning platform.* The Electronic Journal of e-Learning, 4(1), 87-94.

[10] Sodoké, K., Riopel, M., Raîche, G., Nkambou, R. and Lesage, M. (2007). *Extending Moodle functionalities to adaptive testing framework. Paper presented at the World Conference on E-Learning in Corporate, Government*, Healthcare, and Higher Education.

[11] Georgantaki, S. and Retalis, S. (2007). *Using educational tools for teaching object oriented design and programming.* Journal of Information Technology Impact, 7(2), 111-130.

[12] Queirós, R., Oliveira, L., Leal, J. P. and Moreira, F. (2011). *Integration of eportfolios in learning management systems Computational Science and Its Applications*-ICCSA 2011 (pp. 500-510): Springer.

[13] Rößling, G. and Kothe, A. (2009). *Extending Moodle to Better Support Computing Education.* Paper presented at the ITiCSE'09, proceeding of the 14th annual ACM SIGCSE (Innovation and Technology in Computer Science Education), USA, New York.

[14] JISC. (2012). *Joint Information Systems Committee*, 2014, from http://www.jisc.ac.uk/

[15] Deperlioglu, O. and Kose, U. (2013). *The effectiveness and experiences of blended learning approaches to computer programming education.* Computer Applications in Engineering Education, 21(2), 328-342.

[16] Doberkat, E.-E., Engels, G., Hausmann, J. H., Lohmann, M., Pleumann, J. and Schröder, J. (2005). *Software Engineering and eLearning: The MuSofT Project [1]*. Language, 1, 2.

[17] Corbera, F., Gutiérrez, E., Ramos, J., Romero, S. and Trenas, M. A. (2008). *Development of a new MOODLE module for a basic course on computer architecture.* Paper presented at the ACM SIGCSE Bulletin.

[18] Rößling, G. and Vellaramkalayil, T. (2009). *First steps towards a visualization-based computer science hypertextbook as a Moodle module.* Electronic Notes in Theoretical Computer Science, 224, 47-56.

[19] Rodriguez-del-Pino, J. C. (2010). *New Virtual Programming Lab (VPL) module,* 2013, from https://moodle.org/mod/forum/discuss.php?d=154988&username=guest

[20] Prieto-Blazquez, J., Herrera-Joancomartí, J. and Guerrero-Roldán, A.-E. (2009). *A Virtual Laboratory Structure for Developing Programming Labs.* International Journal of Emerging Technologies in Learning.

[21] Fest, A. (2011). *Adding intelligent assessment: a Java framework for integrating dynamic mathematical software components into interactive learning activities*. *ZDM, 43*(3), 413-423.

[22] Lavrishcheva, E., Ostrovski, A. and Radetshiy, I. (2012*). Approach to E-Learning Fundamental Aspects of software Engineering*. (1-10).

[23] Graf, S., & List, B. (2005, July). *An Evaluation of Open Source E-Learning Platforms Stressing Adaptation Issues*. In ICALT (Vol. 5, pp. 163-165).

[24] Al-Khanjari, Z., Al-Roshdi, Y. and Kraiem, N. (2014). *Virt-Res: Developing Extended Architectural Design for Computer Science Virtual Resources Using SOA.* International Journal of Software Engineering and Its Applications, Vol.8, No.9 (2014), pp.125-136. doi: http://dx.doi.org/10.14257/ijseia.2014.8.9.10

[25] Al-Khanjari, Z. and Al-Roshdi, Y. (2014). *Extending the Functionality of LMS to Support Computer Science Education Using Plug-in Tools*. *2014, 6*(2)

[26] Dagger, D., O'Connor, A., Lawless, S., Walsh, E. and Wade, V. P. (2007). *Service-Oriented E-Learning Platforms: From Monolithic Systems to Flexible Services.* Internet Computing, IEEE, 11(3), 28-35. doi: 10.1109/mic.2007.70

[27] Teng, M., Ye, Z. and Junzhou, L. (2005). *A SOA-oriented e-learning resource integration infrastructure and its service discovery model.* Paper presented at the Proceedings of the Ninth International Conference on Computer Supported Cooperative Work in Design

[28] Alario-Hoyos, C., Bote-Lorenzo, M. L., Gómez-Sánchez, E., Asensio-Pérez, J. I., Vega-Gorgojo, G. and Ruiz-Calleja, A. (2012). *GLUE! an architecture for the integration of external tools in virtual learning environments.* Computers & Education.

[29] Moodle. (2013). *About Moodle* Retrieved 16/11/2013, 2013, from http://docs.moodle.org/25/en/Resources

[30] Palumbo, M. and Verga, F. (2013). *Creation of an integrated environment to supply e-learning platforms with Office Automation features.* Interactive learning Environment.

[31] Caballé, S. (2008). *Combining generic programming and service-oriented architectures for the effective and timely development of complex e-learning systems*. Paper presented at the Conference on Complex International and Software Intensive Systems, 2008. CISIS 2008.

[32] Zedan, H. and Al-Ajlan, A. (2007). *E-Learning (MOODLE) Based on Service Oriented Architecture.* Proc. of the EADTU's 20th Anniversary Conference, Lisbon, Portugal, Lisbon-Portugal.

[33] Huertas, F. and Navarro, A. (2013). *Integration Mechanisms in e-learning Platforms.* International Journal of Computer Information Systems and Industrial Management Applications. , Volume 5, 714-721.

Evaluation of Open Source Software with QualiPSO OMM: a case for Bungeni and AT4AM for All

Kennedy Ndenga Malanga [1], Jean Mehat [2], Ivaylo Ganchev [3], John Wandeto [4] and Casper Shivachi Shikali [5]

Abstract - **In this study, we tested the reliability of QualiPSO OMM by using it to evaluate the maturity and quality of Bungeni and AT4AM for All Parliamentary Legislative Systems. We found out that Qualipso OMM assessment has limitations; it relies heavily on product documentation, it is tedious to implement and it does not test code quality effectively. Evaluation and selection of Open Source Software are main challenges affecting adoption of OSS projects. Various methods of evaluating quality of Open Source Software exist, for example Qualipso Open Maturity Model. Unfortunately very few of these methods have been used widely to test OSS projects due to various reasons. We concluded that Qualipso OMM's partial evaluation cannot be relied upon to make decisions on adopting or refraining from an OSS project. Finally, we realized that community metrics can be used to measure quality of OSS projects effectively since such projects are community driven.**

Index terms - **Evaluation, maturity, metrics, Open Source Software, quality.**

XIV. INTRODUCTION

MANY organizations using Free/Open Source Software (FOSS) are dealing with a major problem of selecting the most appropriate software products corresponding to their needs. Open Source Software products targeting the same kind of applications are now very common. This makes choosing an OSS product a tricky task. Quality is one factor that should be considered when choosing among similar open source solutions. The selection process becomes more complicated considering that reliable tools for assessing Open source software quality are rare. It is more challenging to evaluate an OSS project as compared to a proprietary project since the former is usually developed in a dynamic environment. Most organizations select FOSS projects using ad-hoc techniques [1]. Research has shown that practitioners rarely use formal selection procedures. Instead they select OSS products based on familiarity or recommendations by colleagues [4]. Stol and Babar [4] believe that one of the main challenges of using OSS is its evaluation and selection. Different indicators can provide hints on the quality of a FOSS project, for example: the number of users, the longevity of the project, documentation available on-line, number of product downloads and number of bugs reported, etc. All these indicators can have different interpretations. Therefore, it is important to have a structured set of criteria to be used in assessing the quality of OSS projects [2].

A. Methods of Evaluating OSS

Various methods of evaluating quality of Open Source Software have been developed over time which include; Open Source Maturity Model (OSMM) from Cap Gemini, Open Source Maturity Model (OSMM) from Navica, Methodology of Qualification and Selection of Open Source Software (QSOS), Open Business Readiness Rating (OpenBRR), Open Business Quality Rating (Open BQR) and QualiPSO Open Source Maturity Model (OMM). Stol and Babar [4] argue that it is difficult for practitioners to choose a suitable evaluation method. They continue to assert that the key contributing factor to this situation is lack of clarity of the evaluation methods. Petrinja et al. [2] did a comprehensive comparison amongst OpenBRR, QSOS and Qualipso OMM assessment models. From their analysis they concluded that in as much as Qualipso OMM performed as good as OpenBRR and QSOS, it was perceived as being better than the other two. Based on this conclusion, we chose to use Qualipso OMM to evaluate the maturity and quality of Bungeni and AT4AM for All Parliamentary and Legislative Systems (PLS).

B. Objectives

The first objective of this study was to assess the reliability of Qualipso Open Maturity Model of evaluating Open Source Software. Second objective was to establish limitations of this tool when evaluating OSS projects based on partial approach. We were also interested in determining maturity levels of Bungeni and AT4AM for All Parliamentary and Legislative Systems using partial Qualipso OMM evaluation at basic level. Lastly we proposed a better way of evaluating Open Source Software projects.

This study contributes to knowledge in quality of Open Source software since it highlights limitations of using partial Qualipso OMM assessment tool. It also suggests areas that could be focused on when modeling an OSS quality evaluation tool.

[1]University of Paris 8, Laboratoire d'informatique avancée de Saint-Denis (LIASD), 2, rue de la Libert 93526 Saint-Denis Cedex 02, France (phone: +33 1 49 40 64 15 e-mail:liasd@univ-paris8.fr)

[2]Dedan Kimathi University of Technology, School of Computer Science & IT, P.O. Box 657-10100 Nyeri, Kenya (phone: +254 061 2050000 or +254713 835 965 e-mail: info@dkut.ac.ke)

XV. Overview of Qualipso OMM, Bungeni and AT4AM for All

A. Qualipso Open Maturity Model (OMM)

Qualipso Open Maturity Model (OMM) is a process model for OSS development by developers and integration of OSS components by integrators [6]. It is organized as an evolutionary model, inspired by the Capability Maturity Model Integration (CMMI) [6]. Qualipso OMM is organized in three levels i.e. Basic, Intermediate and Advanced levels. Each level builds on the lower levels and also on the Trustworthy elements (TWEs) at the lower level [6]. A trustworthy element (TWE) is a specific factor or aspect of a software development process, or of product results that indirectly influence the perception of the trustworthiness of a FOSS development process. Trust correlates with the quality of a software product and it is influenced by the perception of the quality of trustworthiness of a software product [6]. OMM is based on twelve trustworthy elements as enumerated below [6];

i. Product Documentation (PDOC)
ii. Popularity of the Software Product (REP)
iii. Use of Established and Widespread Standards (STD)
iv. Availability and Use of a (product) Roadmap (RDMP)
v. Quality of Test Plan (QTP)
vi. Relationship between Stakeholders (Users, Developers etc) (STK)
vii. Licenses (LCS)
viii. Technical Environment (Tools, Operating System, Programming Language, Development Environment.) (ENV)
ix. Number of Commits and Bug Reports (DFCT)
x. Maintainability and Stability (MST)
xi. Contribution to FLOSS Product from SW Companies (CONT)
xii. Results of Assessment of the Product by 3rd Party Companies (RASM)

Figure 1 summarizes the TrustWorthy elements in three levels i.e. Basic level, Intermediate level and Advanced level.

To establish the maturity level of an OSS project using Qualipso OMM, it requires an aggregation of the assessment from the practice level up to the maturity level. The following rating algorithms were used to determine the rating of an OSS project at various levels [3];

Rating algorithm of a Practice – R(P);

$$R(P_i) = \frac{\Sigma_{All\ metrics} Metric_i}{number\ of\ Metrics} \quad (1)$$

Rating algorithm of a Goal - R(G);

$$R(G_i) = \frac{\Sigma_{All\ Practices} Practice_i}{number\ of\ Practices} \quad (2)$$

Rating algorithm of a Trustworthy Element - R(TWE);

$$R(TWE_i) = \frac{\Sigma_{All\ Goals} Goal_i}{number\ of\ Goals} \quad (3)$$

Rating algorithm of a Maturity Level - R(ML);

$$R(ML) = \frac{\Sigma_{All\ Practices} P_i}{max \Sigma_{All\ Practices} P_i} \quad (4)$$

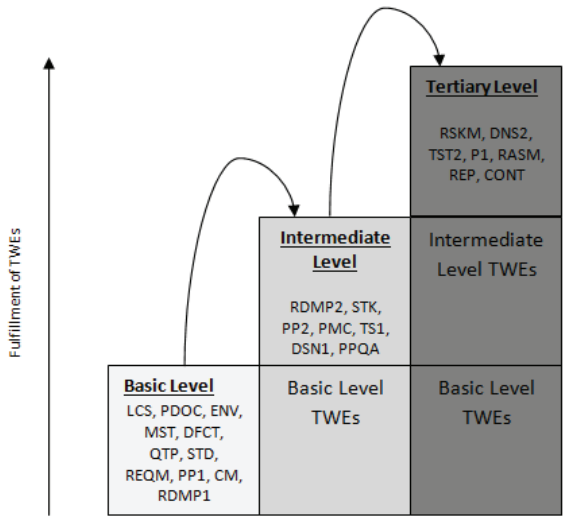


Fig.1: This figure shows three OMM maturity levels. An OSS project qualifies for an Intermediate Level evaluation once all TWEs at Basic Level have been fulfilled. Similarly, Basic Level and Intermediate Level TWEs must be fulfilled for a Tertiary Level evaluation to be carried out.

For an OMM level to be fulfilled, the Maturity Level of the OSS project being assessed should be greater than or equal to 90 percent for that level.

B. Bungeni Parliamentary and Legislative Information System

Bungeni is a system that aims at making parliaments more open and accessible to citizens. This system allows citizens to virtually get inside Parliament" or "Bungeni" the Kiswahili word for "inside Parliament"[8]. Bungeni is an Open Source Software system based on Akoma Ntoso open standards and open source applications. Akoma Ntoso is a framework that defines a set of simple technology neutral electronic representations of parliamentary, legislative and judiciary documents for e-services in a worldwide context and provides an enabling framework for the effective exchange of "machine readable" parliamentary, legislative and judiciary documents such as legislation, debate record, minutes, judgments, etc [7]. Bungeni is an initiative of Africa i-Parliament Action Plan - a programme of United Nations Department of Economic and Social Affairs (UNDESA) [8].

Like many other Open Source software, Bungeni has not been embraced by many parliaments or parliament like organizations. As of today, the stakeholders involved in

Bungeni project include: the national parliaments of Kenya, Mauritius, Mozambique, South Africa, Uganda, Zambia, Nigerian Institute for Legislative Studies, SADC Parliamentary Forum and ECOWAS Parliament [7]. According to Bungeni's project evaluation report of 29th November 2012, this Parliamentary Information System was reported as having proven to be more challenging than had been anticipated. However the report affirms that as of then, the system had matured and that it could be used not just by parliaments but also by other parliament-like organizations such as regional parliaments, municipalities and parliamentary monitoring organizations [7].

C. AT4AM for All Parliamentary and Legislative Information System

AT4AM for All is a free or an open source release of AT4AM. AT4AM is a web-based amendment authoring tool used by the European Parliament to create and table amendments on proposals and present reports of parliamentary committees of the European Commission and the Council of the European Union [12]. The European Parliament decided to use the knowledge and experience gained on the AT4AM project to develop AT4AM for All primarily to help national and regional parliaments to implement their own XML-based amendment authoring systems. Like Bungeni, AT4AM for All uses Akoma Ntoso XML format as input format for source texts and as output format for amendments [12].

Remark: Both Bungeni and AT4AM for All are Parliamentary and Legislative Systems. The reason why these projects were chosen is that they belong to the same family of software. Secondly, most research on OSS focuses on large commonly used projects. Most of the small projects with a high potential like these two projects are mostly overlooked. Despite Bungeni being described as having matured, there is no information that quantifies or describes this maturity. As Deprez and Alexandre point out, organizations want assurance regarding the quality of FOSS projects before integrating them in their solutions [1]. This could explain why adoption of Bungeni is still low. Potential adopters would like to be sure about the level of stability, the degree of support, possibility of adding new features and general continuity of Open Source software before adopting it [5]. We found it necessary to evaluate Bungeni and AT4AM for All quantitatively so as to generate information that potential adopters of these systems can use to gauge the level of maturity of the system before they decide to assimilate them into their work processes.

XVI. METHODOLOGY

A. Approach that was used

As mentioned earlier, Qualipso OMM evaluation is organized in three levels. This study focused on the Basic level of evaluation. To implement this study, a partial OMM assessment was carried out at basic level on Bungeni and AT4AM for All. Qualipso OMM provides for two approaches of assessment. The first approach is a complete assessment of the OSS project, also called internal assessment where the assessment team has got free access to all development documents, process documents and development teams for interview. The second approach is a partial assessment, also known as external OMM assessment, where the assessment team has limited access to the development documentation, process documents and the development team [6]. The first approach is very difficult to achieve in a true Open Source Software development environment since such an environment is usually very dynamic. One reason is that OSS development is usually driven by a community. The developers who contribute to coding are dispersed geographically. They get in and out of the project at different times. Thus it is difficult to assemble the team for purposes of interviews. Secondly, access to documentation is a challenge in a typical OSS development environment. Thirdly, the first approach is feasible in a situation where the project is still under development. However projects at this stage may not be safe to use at industrial level. Projects that have matured most likely have a small support team while the core team will have moved to other projects. As a result, the second approach was chosen for this study since it applies to a majority of OSS projects and that it is the most probable approach small scale potential OSS adopters may choose to quickly evaluate a project.

Remark: Petrinja et al. [2] realized that some aspects of OMM have to be improved so as to enhance its usability. These areas include; identification of misleading questions in QTP (Quality of Test Plan) and ENV (Environment) TrustWorthy Elements, simplifying complexity of questions, simplification of questions' threshold values, automation of scoring mechanism and creation of an easy to understand description of OMM model. We took care of these issues apart from the last two during the development of data collection tool. We found the LookFor for selection of integrated management and communication tool under the Environment (ENV) TrustWorthy Element to be misleading. We realized that a project could have only one environment. As such, all the environments were treated as one group. In addition we did not asses the LookFor on testability testing since we considered testing of testability to be misleading. Any project at any level is testable.

B. Questionnaire

Using the Trustworthy Elements (TWEs) described by Qualipso.org (2014), we designed an assessment questionnaire for the Basic level of the model. We used LimeSurvey to develop an on-line questionnaire for this purpose [11]. Assessment items were based on LookFors and practices. The questionnaire was pilot tested to ensure its reliability and validity. We rated question items using Likert scale of five points where value of 4 corresponded to greater than 75 percent implementation of a practice, value of 3 - a 50 percent to 75 percent implementation of a practice, value of 2 - a 25 percent to 50 percent implementation, value of 1 - less than 25 percent implementation while 0 - for not applicable. To reduce level of bias, the assessment was carried out by participants who were geographically separated. For Bungeni we gathered information from Bungeni portal, Bungeni site and Akoma Ntoso site [10, 8, 7], while for AT4AM for All it we gathered from AT4AM for All site and Atlassian Bitbucket website [12, 9].

C. Data

Table 1 shows data that was collected from this study;

Table 1: TWE scores for Bungeni and AT4AM for All

Trust Worthy Elements (TWE)	Bungeni	AT4AM for All
LCS - Licenses management	2.73	3.12
PDOC - Product Documentation	2.44	1.75
ENV – Environment	3.06	1.31
MST - Maintainability and Stability	2.01	1.02
DFCT - Number of commits and bug reports	1.62	0
QTP - Quality of Test Plan	1.96	1.12
STD - Use of established and widespread Standards	3.19	2.04
REQM - Requirements management	0	0
PP1 - products and Project planning	0	0
CM - Configuration Management	0	0
RDMP – Roadmap	0	0

A total of seventy three practices were assessed for both projects. However out of these practices, data was available for only forty seven practices. The results indicate that both projects scored zero for Requirements management (REQM), Project Planning-1(PP1), Configuration Management (CM) and Roadmap-1 (RDMP) Trustworthy elements. However, AT4AM for All also scored zero for Number of Commits and Bug reports (DFCT) TWE.

XVII. Results and Analysis

Results show that Bungeni scored better in most of the TWEs as compared to AT4AM for All. Figure 2 summarizes this comparison;

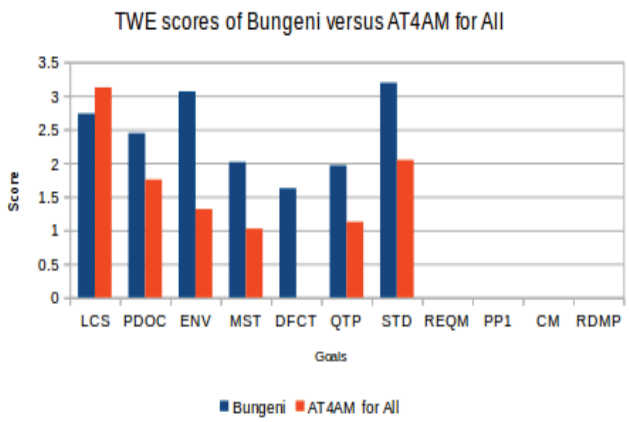


Fig. 2: A comparison graph of TWE scores of Bungeni and AT4AM for All

An analysis of fulfilled Trustworthy elements for each project was done as depicted by the graphs in figure 3 and figure 4.

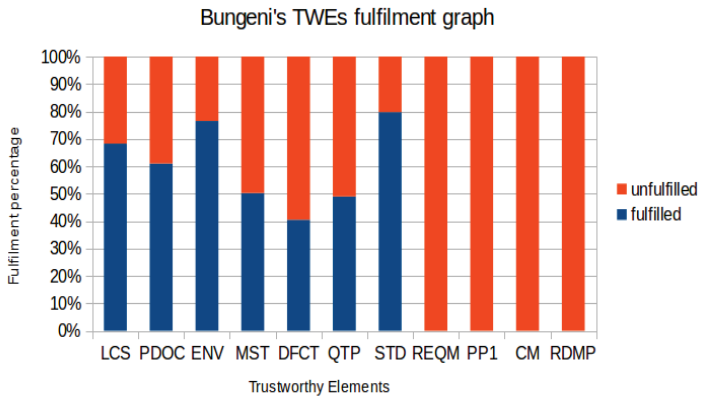


Fig. 3: Bungeni's Trustworthy Elements (TWEs) fulfillment graph

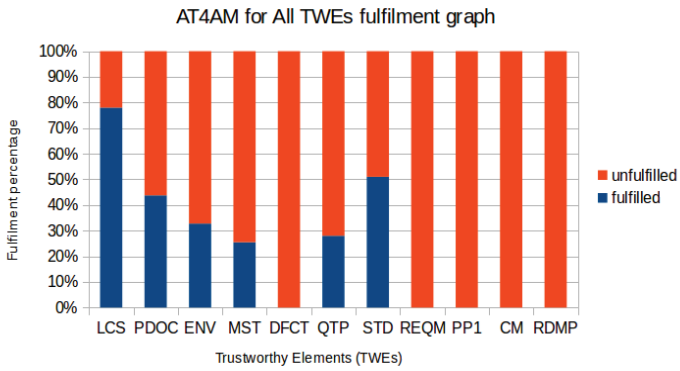


Fig. 4: AT4AM for All's Trustworthy Elements (TWEs) fulfillment graph

Using equation 4, we determined the Maturity level for the two projects. For all seventy three practices under Basic level of OMM, Bungeni scored a Maturity Level of 40.08 percent while AT4AM for All scored a Maturity Level of 29.28 percent. Qualipso OMM gives a leeway for a partial assessment not to evaluate all practices. The assessors are given freedom to select the practices to be considered for assessment [6]. Using this rationale we selected forty seven practices whose information was accessible over the internet. Subsequently we determined a Maturity Level for each project. For this scenario Bungeni scored 62.25 percent while AT4AM for All scored 45.47 percent.

A. Discussion

Partial Assessment: For partial assessment of OSS projects, the assessor does not get full insight into the project but only gets an outside view [6]. This statement raises many questions on the understanding of what an assessment is. An assessment includes all the various methods used to determine the extent of an achievement of a standard. It is very difficult to measure outside view of an item since viewing something is very subjective. Qualipso has set a standard of greater than or equal to 90 percent for an OSS project to qualify to be considered as trustworthy and of high quality. However this standard applies only for full assessment. For every kind of assessment, a standard must be involved so as to act as a reference point when making decisions. However, Qualipso OMM does not define the minimum standard for partial assessment. Since partial assessment should be able to assist a potential adopter to

make a decision for either adopting or rejecting an OSS project, there is need for some level of a minimum standard. The task of setting a standard is left to the assessors. This situation introduces high degree of subjectivity that ends up influencing the validity and reliability of the results. In a nutshell, construct validity is negatively affected since the scores achieved from partial assessment may not necessarily reflect the items being tested. For example considering the scores of Bungeni and AT4AM for All as shown in the previous section, these projects are depicted as being not mature. It may appear as if the projects have not reached a usable maturity level. Actually this is not the case. Both projects are already operational. The fact is that partial Qualipso OMM assessment does not provide valid information that potential adopters can rely on to make decisions to either adopt or refrain from a specific OSS project. However the model can be enhanced by introducing a standard of some minimum requirements that can be measured reliably for partial assessment.

The role of documentation: Evaluation criterion for Qualipso OMM is highly dependent on documentation. This model will only guarantee reliable results in situations where documentation is available. Developers working on small scale non-funded OSS projects tend to focus more on the end product rather than documentation. If such projects are assessed by Qualipso OMM, definitely they will score poorly and yet the end product could be of good quality.

The role of the development team: At some point during evaluation process using Qualipso OMM, it becomes very necessary to be in contact with a person from the development team. This is true since sixty seven percent of LookFors at the Basic level of this model may require clarification from the developer. The availability of the developer may not be guaranteed considering the dynamic nature of OSS development. For example due to code contribution, there could be no distinct owner of the project, or the developer may have stopped supporting the project. In this situation then Qualipso OMM will deliver a biased evaluation as it depends on the project's scanty documentation found on the Internet.

Satisfaction level of users: The practices on use of Established Standards (STDs) and Number of Commits and bugs (DFCT) within Qualipso OMM focus on assessing whether mechanisms for measuring satisfaction level of users have been implemented by the project. We envisaged that, user satisfaction is a very significant element in evaluating quality of software. Reza et al. maintain that user satisfaction has always been a major factor in the success of software, regardless of whether it is closed proprietary or OSS [19]. Therefore it would have been prudent for Qualipso OMM to assess the satisfaction level of users directly from users of the project being assessed instead of inferring from the developer. As mentioned earlier many OSS projects may not have the luxury to carry out elaborate surveys to assess the usability of their products.

Process versus product assessment: Qualipso OMM emphasizes the assessment of the development process of an OSS project. Zieliski and Szmuc argue that successful projects employ more mature processes than unsuccessful projects [13]. However this may not necessarily always be the case. In as much as the process of development may be an indicator of quality, the ultimate quality is defined by the final product. Very little assessment of quality of the code for the product has been carried out by this method. It is very necessary to measure the quality of code so as to enable potential integrators understand how easy or difficult it is to integrate or reuse the code of an OSS project.

FLOSS Community metrics: Finally, Open Source Software community metrics can be used as an indicator of quality of an OSS project. Falkner [14] has defined second order FLOSS community metrics which include; crowd sourced quality, time between contribution, location of community members, evolution of contributor age, evolution of active or inactive user, time of bug report, time of bug being fixed, number of ignored contributions and distribution of commits across functional areas. If such FLOSS metrics are well understood, they can be used as a pointer towards quality of an Open Source project. Many tools are upcoming that will enable mining of data from on-line repositories for the FLOSS Community Metrics. Such tools include; MetricsGrimoire, Riscoss Analytics, OSSMETER and Eclipse Community Metrics [15 - 18]. Using the output from these tools, potential OSS adopters will be able to easily establish indicators of quality of an OSS project with less effort.

XVIII. Conclusion

Qualipso OMM is a good tool for evaluating Open Source Software. However it can be rendered unreliable if project documentation is not available. Qualipso OMMs partial assessment cannot be relied on when making decisions of either adopting or refraining from a particular Open Source Software project. The ultimate quality of a software project is defined in terms of the code quality and its usability. Qualipso OMM has not explored these two areas in details and yet they are very important. Generally, Qualipso OMM is bound to be effective in an environment where Open Source Software project is developed following a proprietary-like work-flow.

We should therefore build a better Open Source Software assessment tool that can automate the process of evaluation. For example an on-line evaluation tool that can automatically mine for data from repositories and generate maturity information of OSS projects. The tool should also be able to compare two or more similar OSS projects statistically in terms of quality. Finally such a tool should be reliable with an ability to predict the life expectancy of an OSS project in relation to support from the community.

Future work: Our future work entails developing a model of evaluating quality of Open Source Software projects based on Open Source Software community metrics. Such a model should be able to determine the quality of past, current, and future versions of an open source software product.

Acknowledgment

We sincerely thank the Advanced Computing Laboratory of Saint-Denis (LIASD) for supporting this study.

REFERENCES

[1]. Deprez, J.-C. & Alexandre, S. (2008). Comparing assessment methodologies for free/open source software: OpenBRR and QSOS. Product-Focused Software Process Improvement (pp. 189203). Springer.

[2]. Petrinja, E., Sillitti, A. and Succi, G. (2010). Comparing OpenBRR, QSOS, and OMM assessment models. Open Source Software: New Horizons (pp. 224238). Springer.

[3]. Petrinja, E. and Succi, G. (2012). Assessing the open source development processes using OMM. Advances in Software Engineering, 2012, 8. Hindawi Publishing Corp.

[4]. Stol, K.-J. and Babar, M. A. (2010). A comparison framework for open source software evaluation methods. Open Source Software: New Horizons (pp. 389394). Springer.

[5]. QSOS core team et al. (2013). Collaborative technological watch: Qualification and Selection of Opensource Software. Retrieved Sept 13th, 2014, from QSOS: http://www.qsos.org/

[6]. Qualipso.org. (2014). Retrieved August 2nd August 2014, 2014, from Qualipso: Quality platform for Open Source Software: http://qualipso.icmc.usp.br/OMM/

[7]. UNDESA - Akoma Ntoso. (2014). Project Evaluation. Retrieved August 8th August 2014, 2014, from Akoma Ntoso: www.akomantoso.org

[8]. UNDESA. (2014). Bungeni. Retrieved August 1, 2014, from Bungeni: http://www.bungeni.org/

[9]. Atlassian. (2014). AT4AM code. Retrieved from Atlassian Bucket: http://code.at4am.org/

[10]. UNDESA Bungeni-Portal. (2014). Project Home. Retrieved from Bungeni-Portal: https://code.google.com/p/bungeni-portal

[11]. LimeSurvey. (2014, October 2014 27th). Home. Retrieved from LimeSurvey: https://www.limesurvey.org/en

[12]. European Parliament. (2014, October 27th). About. Retrieved from AT4AM for All: http://www.at4am.org/overview/

[13]. Zieliski, K. and Szmuc, T. (2005). Software process maturity and the success of free software projects. Frontiers in Artificial Intelligence and Applications, 3.

[14]. Falkner, J. (2014, November 2). Metrics Are Fun, but which ones really matter? Retrieved from Speaker Deck: https://speakerdeck.com/ossmetrics/metrics-are-fun-but-which-ones-really-matter

[15]. GitHub . (2014, November 2nd). MetricsGrimoire. Retrieved from MetricsGrimoire - Tools for software and development community analytics: http://metricsgrimoire.github.io/

[16]. Milinkovitch, M. (2014, November 2nd). Eclipse Community Metrics. Retrieved from Speaker Deck: https://speakerdeck.com/ossmetrics/eclipse-community-metrics

[17]. OSSMETER Project. (2014, November 2nd). Main. Retrieved from OSS-METER - Automated Measurement and Analysis of Open Source Software: http://www.ossmeter.org

[18]. OW2. (2014, November 2). Main. Retrieved from riscoss - managing risk and cost in open source software: http://www.riscoss.eu/bin/view/Main/

[19]. Reza, A., Capretz, L. F. & Ahmed, F. (2012). An open source usability maturity model (OSUMM). Computers in Human Behavior, 28(4), 1109–1121.

Open-Source Linguistic Techniques for Knowledge Discovery Processing

Khaled Abdalgader[1], Mohamed S. Hajji[2]

***Abstract*—Open-source based development provides, in comparison with traditional development methods, foundation code which developers can experiment with and modify according to their experimental needs. This is important in knowledge discovery processing, since it could help researchers to cope with the enormous flexibility of natural language which enables humans to write similar meanings using quite different language structures. In addition to describing computational linguistic techniques relevant to knowledge discovery processing, this paper introduces how the work was developed as open source modules so it can be used for experimental purposes. The open-source developed techniques for knowledge discovery processing include word sense disambiguation step, short-text similarity measurement step, and text clustering step. The concluding remarks discuss these techniques and the benefits of being open-source and show that they are not independent, as the performance of one often depends on the performance of another.**

***Index Terms*— Knowledge Discovery, Short-Text Similarity Measurement, Text Clustering, Open Source Code.**

XIX. INTRODUCTION

THe availability of large text collections stored in electronic repositories has created the potential of a vast amount of valuable knowledge buried in those texts. This in turn has created the need for automated techniques for deriving relevant and useful knowledge. Knowledge discovery (i.e., text mining) can broadly be described as the process of deriving high quality information from text, where 'high quality' refers to some combination of relevance, novelty, and interestingness [1].

Tasks performed in knowledge discovery applications include, typically, text categorization (i.e., classifying a text fragment as belonging to one or more predefined classes or categories) [2]; text clustering (i.e., grouping text fragments according to their degree of similarity to one another) [3, 4], and text summarisation (i.e., producing a document summary which captures the main body of relevant content in some document or documents) [5, 6]. These tasks are not independent, an activity focused on text summarisation, for example, may involve sub-tasks involving classification or clustering [7].

Various authors have argued that using open-source tools for knowledge discovery processing helps to avoid problems of inflexibility and to modify functions to fit some needs of the experimental development process. Additionally, the problem of incompatible formats that appear in traditional tools can be avoided. This is important in knowledge discovery processing because of the need to handle the flexibility of natural language where similar meaning can be expressed in a wide range of language structures [8].

Consequently, the availability of open-source subsystems might help researchers to concentrate on experimenting and/or developing some tasks of the discovery process while using ready open-source for other tasks or even for the same task they are working on. Since, while conducting this work, no satisfactory open-source code was found, it was decided to build different parts of the system as open-source components.

This paper introduces the founded computational linguistic techniques relevant to knowledge discovery processing. These include word sense disambiguation, which usually operate at the word level to identify the actual meaning of the word [9-11]; text similarity measurement, where it is necessary to calculate the similarity value between a short-text pair [12-15]; and text clustering, where we are interested in the grouping text fragments according to their degree of similarity [16, 17]. The work was based on developing separate components of open-source code each of which support one of the techniques mentioned above. The availability of such open-source components, or more precisely the lack of it, is discussed too.

The remainder of the paper is structured as follows. Section II starts with a brief introduction to the levels of linguistic processing. Section III presents the developed open-source algorithm to word sense disambiguation (WSD). Section IV introduces the developed open-source sentence-level text similarity method. Developed open-source clustering algorithm that can be used to cluster short-text is presented in Section V. Section VI concludes the paper by presenting a detailed discussion of the developed computational linguistic techniques and the benefits of being open-source and show that they are not independent, as the performance of one often depends on the performance of another.

XX. LEVELS OF LINGUISTIC PROCESSING

As mentioned in the previous section, knowledge discovery processing include text categorization, text clustering and text summarisation. These tasks depend on each other. Consider, for example, the problem of document summarisation [18, 19]. One approach to document summarisation is to identify the main themes or topics which characterise a document, and to then construct a summary of the document by appending, in a coherent

[1]Faculty of Computing and Information Technology, Sohar University, P.O. Box: 44, P. Code 311, Sohar, Sultanate of Oman (phone: +968-26720101; fax: 968-26720102; e-mail: komar@soharuni.edu.om)

[2]Faculty of Computing and Information Technology, Sohar University, P.O. Box: 44, P. Code 311, Sohar, Sultanate of Oman (phone: +968-26720101; fax: 968-26720102; e-mail: mhajji@soharuni.edu.om)

manner, a description of each of those themes. Presumably, fragments of text that are similar to each other are more likely to relate to the same theme than fragments that are less similar. Thus, clustering, using both an appropriate similarity measure and an appropriate level of text fragmentation should provide a useful tool in allowing us to identify those themes.

An important question, then, is what unit of text should be used for clustering on tasks such as text summarisation: words, phrases, sentences, paragraphs, etc.? If the unit of fragmentation is too small (e.g., individual words), we may succeed in finding clusters of related words, but it will be difficult to recombine these words to create a summary. On the other hand, if the unit is too large (e.g., a paragraph or document), then we may not be able to clearly identify themes, since a paragraph may span a number of topics. Sentences are probably at about the right level of fragmentation since they tend to contain information about specific events, and are therefore more likely to provide a suitable context for identifying themes [20, 21].

A second important question concerns representation; i.e., how should sentence-level text be represented in order that an appropriate similarity measure can be defined? Representations such as the Vector Space Model (VSM) [22], which are based solely on word co-occurrence and commonly used at the document level, are clearly not suitable at the sentence level, since two sentences may be about a similar topic, yet contain no words in common. Thus, at the sentence-level, we require a representation which is better able to capture the *semantic* content of sentences, thereby enabling a more appropriate similarity measure to be defined.

XXI. Word Sense Disambiguation

Word sense disambiguation (WSD)—the process of identifying the appropriate meanings of words as they occur in a sentence—is an intermediate and fundamental task in many natural language processing applications. In this section, we first review the related approaches to WSD and then briefly introduce our developed open-source WSD algorithm.

A. *Word Sense Disambiguation Methods*

Various WSD methods have been proposed in the literature [9-11], and can be broadly classified as belonging to one of two families: *corpus-based* approaches, and *knowledge-based* approaches.

Corpus-based methods utilise *supervised* learning techniques to induce a classifier from a corpus of training data consisting of a set of labeled words, in which the label indicates the sense in which the word is being used. Once a classifier has been created by extracting the syntactic and semantic features, it can then be used to predict the sense of the target word in novel sentences. In contrast, knowledge-based methods are usually *unsupervised* and do not require any such corpora, relying instead on external lexical resources such as dictionaries or thesauri [23].

Knowledge-based WSD methods fall into two main groups: *similarity-based* methods, and *graph-based* methods. Similarity-based methods determine the sense of a polysemous word by computing the similarity between each of its possible senses and the words in the surrounding context. The correct sense of the target word is then assumed to be that for which the similarity is greatest. Graph-based methods, however, usually build a semantic structure (i.e., a graph) representing all available senses of all of the words being disambiguated. The nodes in this graph correspond to these senses and the edges represent the lexical relations between them. Graph centrality methods are then typically used to determine which nodes are more important (i.e., central) within the graph, and these are considered to be the correct senses of the target words. Because they disambiguate all words in a text fragment simultaneously by exploiting semantic similarities across word senses, graph-based methods usually achieve higher performance than similarity-based methods, which disambiguate words individually, usually without considering the senses assigned to surrounding words [9, 10]. However, the main disadvantage of graphical methods is their high computational complexity.

As a one of the developed open-source linguistic technique introduced in this paper, Abdalgader and Skabar (2012) [9] proposed a new unsupervised similarity-based WSD algorithm that operates by computing the semantic similarity between glosses of the target word and a context vector. The sense of the target word is determined as that for which the similarity between gloss and context vector is highest. This WSD algorithm has previously been reported in [9].

B. *Evaluation of Word Sense Disambiguation Methods*

WSD methods are often evaluated using a different models of evaluation (i.e., *in vitro* and *in vivo*) [9][24] and for *in vitro* model, there is a several standard datasets have been constructed specifically for this purpose. For example, the Senseval/Semeval[3] campaign provides a shared task with a variety of datasets and sense inventories for all-words and lexical sample settings in different languages. The SemCor [25], Senseval-2 [26] and Senseval-3 [27] datasets are the most common standard datasets.

XXII. Short-Text Similarity Measurement

The vector space model has been successful in IR because it is able to adequately capture much of the semantic content of document-level text. This is because documents that are semantically related are likely to contain many words in common, and thus are found to be similar according to popular vector space measures, which are based on word co-occurrence [28]. However, while the assumption that (semantic) similarity can be measured in terms of word co-occurrence may be valid at the document level, the assumption does not hold for small-sized text fragments such as sentences, since two sentences may be semantically related despite having few, if any, words in common.

Various linguistic measures for sentence similarity have been proposed in recent years [12-15]. Rather than representing sentences in a common vector space, most of these measures represent the sentences in a reduced vector space consisting only of the words contained in the

[3] A series of international WSD competitions (http://www.senseval.org), organized by the ACL-SIGLEX that has been held every three years since 1998.

sentences, and define similarity as some function of inter-sentence word-to-word similarities, where these similarities are in turn usually derived either from distributional information from some corpora (corpus-based measures), or semantic information represented in external lexical sources (knowledge-based measures).

Unlike existing measures, which use the set of exact words that appear in the sentences, Abdalgader and Skabar (2011) [12] proposed a novel method (which is one of the developed open-source linguistic technique described in this paper) constructs an expansion word set for each sentence using synonyms of the disambiguated words in that sentence. This open-source sentence similarity measure has previously been reported in [12].

A. *Evaluation Short-Text Similarity Measures*

Sentence similarity measures are commonly evaluated as *in vitro* task. To this end, a variety of sentences datasets have been constructed, and include the Microsoft Research Paraphrase (MSRP) Corpus [29], and the Recognising Textual Entailment (RTE) challenge dataset [30]. Each of these datasets is structured as a collection of sentence pairs, where each pair is tagged with a binary class value of 1 or 0, indicating whether the sentences are similar or dissimilar. The problem is thus a binary classification task in which the objective is to correctly predict the class membership of the sentence pair, with performance measured using standard binary classification measures such as accuracy, precision, recall and F-measure [31].

One of the difficulties in measuring the performance on *in vitro* tasks such as these is that a classification threshold must be determined, and the performance can be very sensitive to choice of this threshold. Most researchers simply use a threshold of 0.5, although some authors have experimented with other threshold values [15].

The need to determine a threshold can also be avoided by using *in vivo* evaluation model. For example, the task of clustering relies on a measure of similarity between the objects (sentences) being clustered and the quality of clustering can be taken as being indicative of the sentence similarity measure being used.

XXIII. Short-Text Clustering

Short-text clustering plays an important role in many text mining activities. For example consider web mining [32], where the specific objective might be to discover some novel information from a set of documents initially retrieved in response to some query. By clustering the sentences of those documents we would intuitively expect at least one of the clusters to be closely related to the concepts described by the query terms; however, other clusters may contain information pertaining to the query in some way hitherto unknown to us, and in such a case we would have successfully mined new information.

Irrespective of the specific task (e.g., summarisation, text mining, etc.), most documents will contain interrelated topics or themes, and many sentences will be related to some degree to a number of these. This means that ideally a clustering algorithm should be able to identify soft or *fuzzy* clusters [16], in which sentences belong to all of these clusters with different degrees of membership. However, clustering text at the sentence level poses specific challenges not present when clustering larger segments of text, such as documents.

A. *Fuzzy Clustering Approaches*

A very recent open-source algorithm for fuzzy relational clustering at short-text level is proposed by Skabar and Abdalgader 2013 [16]. It is based on expectation maximization; however, unlike conventional methods which represent clusters using Gaussian components, the proposed algorithm is based on estimating the likelihood of an object using the well-known PageRank algorithm [33]. Then, it allows patterns to belong to all clusters with differing degrees of membership. This is important in domains such as sentence clustering, since a sentence is likely to be related to more than one theme or topic present within a document or set of documents. Figure 1 shows fuzzy membership for a resulted cluster. The figure also shows global weighted PageRank scores. Cluster centroids (red) are distributed around the denser inner region of the graph.

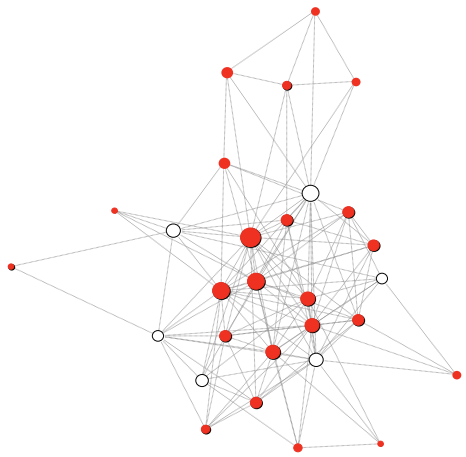


Fig. 1 a fuzzy cluster membership for produced cluster of our developed fuzzy sentence clustering algorithm. Size of nodes represents the degree of membership of a sentence to the cluster. Nodes in (red) are cluster centroids.

XXIV. Discussion and Concluding Remarks

This paper has introduced the developed open-source computational linguistic techniques for knowledge discovery that related to the areas of word sense disambiguation, sentence similarity measurement, and sentence clustering. It is important to stress that these knowledge discovery activities are not independent, and the performance of one often depends on the performance of another. For example, clustering sentences relies on a measure of sentence similarity; a good measure of sentence similarity should be able to effectively utilise the context; effectively using context will rely on an ability to identify the correct sense of words, and so on. Figure 2 shows the relationship between these open-source linguistic techniques for knowledge discovery processing.

The research described in this paper was motivated by the belief that being able to capture such semantic relationships through using open-source linguistic techniques will lead to an increase in the scope breadth of

problems to which clustering, classification, and other knowledge discovery processing can be applied at the sentence level. However, the performance of any clustering or classification algorithm will be limited by the quality of the input data, and in the case of sentence-level text, performance will depend fundamentally on the quality of the sentence similarity measure that is used.

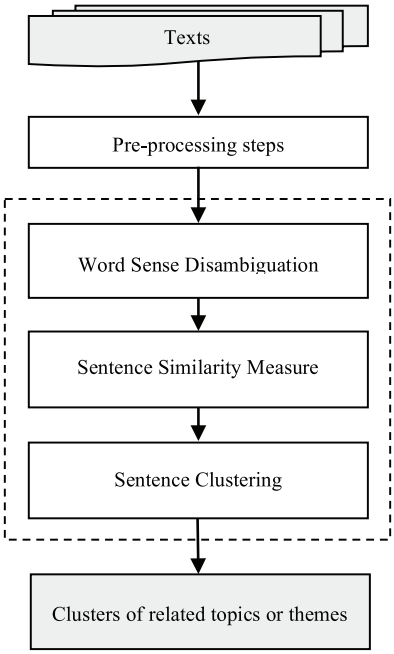


Fig. 2 a relationship between developed open-source computational linguistic techniques for knowledge discovery processing.

Different literatures, describe open-source code for different tasks of knowledge discovery processing. To the best of our knowledge, however, there is no proper open source code relevant to this field of research. For example in BabelNet publications like [34], we did not find any mentioning of open-source work. On the other hand, BableNet provides API tools for multilingual lexical semantic analysis [35]. This tool, though it is helpful, is far from the open source components advocated in this work.

According to developers, MEAD [36] can perform summarization to multiple documents in different languages. It is a public open-source environment used by several implementers for some web applications as well as mobile apps. MEAD is, nevertheless, more concerned with document level processing and does not satisfy this research interests. It is not meant to fit sentence level or to cover related techniques. MEAD framework cannot be scaled down to work with the level of our concerns.

The open-source developed in this work was written in Python with Natural Language Toolkit, the commonly used programming languages in the field of this research, let alone it is becoming very popular in general. The code covers the three main tasks that support text knowledge discovery. It is not completely finalised but was ready for colleagues in our faculty to experiment with. Some colleagues used and modified the code to implement their algorithms concerning different areas of knowledge discovery like similarity measurements and clustering and to check the performance of their ideas. This open-source code proved very helpful and time saving let alone it provided flexibility to researcher. The work of colleagues was a chance also to enhance the open-source code and consolidate its role. Figure 3 shows a portion of our developed open-source linguistic techniques for knowledge discovery processing as implemented in Python environment.

Fig. 3 a portion of the developed open-source linguistic techniques for knowledge discovery processing as implemented in Python.

In regard to word sense disambiguation, however, it has been shown that graph-based methods usually achieve higher performance than their similarity-based alternatives. This is because they disambiguate all words in a text fragment simultaneously, whereas similarity-based methods disambiguate words individually, usually without considering the senses assigned to surrounding words. The main disadvantage of graphical methods is their high computational complexity. Although similarity-based methods are usually far more efficient than graph-based methods, they are usually based either on gloss-context word overlap [11] or measuring pairwise similarity between word-senses [37], and do not fully utilise the semantic information associated with word-senses that is available through resources such as WordNet's glosses [38]. Also, due to computational requirements, they are usually limited to using context from only a small window surrounding the target word.

Most of the proposed sentence similarity measures compute similarity between their constituent words based either on distributional information from some corpora (corpus-based measures), or on semantic information represented in external sources such as WordNet [38] (knowledge-based measures). Some of the measures, including those of Li *et al.* (2006) [14] and Mihalcea *et al.* (2006)[15], incorporate a measure based on the word's importance, as measured, for example, by the word's IDF score, or *information content*, in the case of Li *et al.* (2006)[14]. The rationale for this is that words which have a higher IDF are more important, and thus should contribute more heavily in the sentence similarity calculation than less important words. However, while it is widely accepted that incorporating IDF scores leads to improved measurement of text similarity at the document level, it is not clear that it has the same utility at the sentence level. Another difficulty in using IDF scores at the sentence level is that many words are polysemous (i.e., have multiple meanings). Even if the sense of

these words can be determined, there remains the problem that IDF scores are generally not available for specific senses of words. Skabar and Abdalgader , 2011 [13] tackles these issue by exploring the idea of incorporating into sentence similarity methods a factor based on the importance of words in the actual sentences being compared (as opposed to average importance over some large corpus).

While a number of papers have reported on the development and evaluation of sentence similarity measures [14, 15, 39], most of these are based on word-to-word similarity using the first sense of each of the words being compared, and intuitively, identifying the correct sense in which a word is being used should lead to a more accurate measure of the similarity between two sentences. To date there has been very little research reporting the incorporation of WSD into sentence similarity measurement. The exceptions are Abdalgader and Skabar (2011)[2] and Ho *et al* (2010)[40].

In examining the literature on clustering, we have focused on fuzzy relational clustering. This was motivated by the fact that: (i) widely used sentence similarity measures such as those of Li *et al*., (2006) [15] and Mihalcea *et al*., (2006)[15] do not represent sentences in a common metric space, thereby requiring a relational, as opposed to attribute-based, approach to clustering, and (ii) sentences are unlikely to relate to just a single concept or theme within a document, but to a number of themes. There are currently very few algorithms falling into this category. Skabar and Abdalgader (2013) [16] presented a new fuzzy clustering algorithm which operates on relational input data; that is, data in the form of a square matrix of pairwise similarities between sentences.

Open-source linguistic techniques are an exciting area of research within the knowledge discovery community, and this paper has introduced several of those techniques which are able to identify word senses, sentence similarity and fuzzy clusters based on relational input data. We have already mentioned some of the new work we are conducting in this area; however, what we are most excited about is extending the technique to perform opinion mining. The concepts present in natural language documents usually display some type of people opinion, whereas the techniques we have presented in this paper identifies only themes or topics. Our main future objective is to extend these ideas to the development of complete open-source linguistic techniques for sentiment analysis or opinion mining.

REFERENCES

[1] Hotho, A., Nürnberger, A. and Paaß, G. 2005. A brief survey of text mining. *GLDV-Journal for Computational Linguistics and Language Technology*. 20, 19-62.

[2] Namburu, S.M., Tu, H., Luo, J., and Pattipati, K.R. 2005. Experiments on supervised learning algorithms for text categorization. *IEEE Aerospace Conference*. Big Sky, MT.

[3] Hatzivassiloglou, V., Klavans, J. L., Holcombe, M. L., Barzilay, R., Kan, M.-Y. and McKeown, K. R. 2001. SIMFINDER: A flexible clustering tool for summarization. In *NAACL Workshop on Automatic Summarization*. Association for Computational Linguistics. 41-49.

[4] Radev, D.R., Jing, H., Stys, M., and Tam, D. 2004. Centroid-based summarization of multiple documents. *Information Processing and Management: AN International Journal*. 40, 919-938.

[5] Chen, F., Han, K. and Chen, G. 2008. An Approach to sentence selection based text summarization. In *Proceedings of IEEE TENCON02*. 489-493.

[6] Kyoomarsi, F., Khosravi, H., Eslami, E., Dehkordy, P.K and Tajoddin, A. 2008. Optimizing text summarization based on fuzzy logic. *Seventh IEEE/ACIS International Conference on Computer and Information Science, IEEE Computer Society*. 347-352.

[7] Vidhya, K. A. and G. Aghila, G. 2010. Text Mining Process, Techniques and Tools: an Overview. *International Journal of Information Technology and Knowledge Management*. 2, 2, 613-622.

[8] Winograd, T. (1972). Understanding natural language. *Academic Press*, Inc. Orlando, FL, USA.

[9] Abdalgader, K. and Skabar, A. 2012. Unsupervised similarity-based word sense disambiguation using context vectors and sentential word importance. *ACM Transactions on Speech and Language Processing*.

[10] Navigli, R. and Lapata, M. 2010. An experimental study of graph connectivity for unsupervised word sense disambiguation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 32, 4, 678-692.

[11] Lesk, M. 1986. Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In *Proceedings of the 5th annual international conference on systems documentation*. (SIGDOC, Toronto, Canada). 24-26.

[12] Abdalgader, K. and Skabar, A. 2011. Short-text similarity measurement using word sense disambiguation and synonym expansion. In *Proceedings of the* 23rd Australasian Joint Conference on Artificial Intelligence. (AI2010, Adelaide). Advances in Artificial Intelligence. 6464, 435-444.

[13] Skabar, A. and Abdalgader, K. 2011. Improving sentence similarity measurement by incorporating sentential word importance. In *Proceedings of the* 23rd Australasian Joint Conference on Artificial Intelligence. (AI2010, Adelaide). Advances in Artificial Intelligence. 6464, 466-475.

[14] Li, Y., McLean, D., Bandar, Z., O'Shea, F. and Crockett, K. 2006. Sentence similarity based on semantic nets and corpus statistics. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*. 18, 8, 1138-1150.

[15] Mihalcea, R., Corley, C., Strapparava, C. 2006. Corpus-based and knowledge-based measures of text semantic similarity. In *Proceedings of 21st National Conference on Artificial Intelligence*. 1, 775-780.

[16] Skabar, A. and Abdalgader, K. 2013. Clustering sentence-level text using a novel fuzzy relational clustering algorithm. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*. IEEE computer Society Digital Library. IEEE Computer Society.

[17] MacQueen, J. B. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*. 281-297.

[18] Zha, H. 2002. Generic summarization and keyphrase extraction using mutual reinforcement principle and sentence clustering. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*. 113-120.

[19] Aliguyev, R.M. 2009. A New sentence similarity measure and sentence based extractive technique for automatic text summarization. *Expert Systems with Applications*. 36, 7764-7772.

[20] Pedersen, T. 2008. Computational approaches to measuring the similarity of short contexts: a review of applications and methods. CoRR. Abs/0806.3787.

[21] Naughton, M., Kushmerick, N. and Carthy, J. 2006. Clustering sentences for discovering events in news articles. In *Proceedings of ECIR*. 535-538.

[22] Salton, G. 1989. Automatic text processing: The transformation, analysis, and retrieval of information by computer. Addison-Wesley, Reading, Mass.

[23] Navigli, R. 2009. Word sense disambiguation: A survey. *ACM Computing Surveys*. (CSUR). 41, 2, 1-69.

[24] Kilgarriff, A. and Palmer, M. 2000. Introduction to the special issue on Senseval. *Comput. Human*. 34, 1-2, 1–13.

[25] Miller, G., Leacock, C., Randee, T., and Bunker, R. 1993. A semantic concordance. In *Proceedings of the 3rd DARPA workshop on Human Language Technology*. 303-308.

[26] Palmer, M., Fellbaum, C., Cotton, S., Delfs, L., and Dang, H. 2001. English tasks: all-words and verb lexical sample. In *Proceedings of ACL/SIGLEX* (Senseval-2, Toulouse, France). 21–24.

[27] Snyder, B. and Palmer, M. 2004. The English all-words task. In *Proceedings of ACL/SIGLEX*. (Senseval-3, Barcelona, Spain). 41–43.

[28] Manning, C. D., Raghavan, P., Schütze, H. 2008. *Introduction to information retrieval*. Cambridge University Press, Cambridge.

[29] Dolan, W, Chris Quirk, C, and Brockett, CV. 2004. Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In *Proceedings of the 20th International Conference on Computational Linguistics*. 350-356.

[30] Dagan, I, Glickman, O. and Magnini, B. 2005. The PASCAL recognizing textual entailment challenge. In *Quinonero-Candela et al*., editor, MLCW 2005. Springer-Verlag, LNAI, 3944, 177–190.

[31] Van Rijisbergen, C.J. 1975. *Information Retrieval*. Butterworths, London.

[32] Kosala, R. and Blockeel, H. 2000. Web mining research: A survey. *ACM SIGKDD Explorations Newsletter*. 2, 1, 1-15.

[33] S. Brin and L. Page. 1998. The Anatomy of a Large-Scale Hypertextual Web Search Engine. *Computer Networks and ISDN Systems*, vol. 30, pp. 107-117.

[34] R. Navigli, D. A. Jurgens, D. Vannella. SemEval-2013 Task 12: Multilingual Word Sense Disambiguation. Proc. of 7th International Workshop on Semantic Evaluation (SemEval), in the Second Joint Conference on Lexical and Computational Semantics (*SEM 2013), Atlanta, USA, June 14-15th, 2013, pp. 222-231

[35] R. Navigli and S. Ponzetto. Multilingual WSD with Just a Few Lines of Code: the BabelNet API. *Proc. of the 50th Annual Meeting of the Association for Computational Linguistics* (ACL 2012), Jeju, Korea, July 9-11, 2012, pp. 67-72.

[36] Dragomir Radev; Timothy Allison; Sasha Blair-Goldensohn; John Blitzer; Arda Celebi; Stanko Dimitrov; Elliott Drabek; Ali Hakim; Wai Lam; Danyu Liu; Jahna Otterbacher; Hong Qi; Horacio Saggion; Simone Teufel; Michael Topper; Adam Winkel; Zhu Zhang. MEAD - A Platform for Multidocument Multilingual Text Summarization. *Proceedings of the 4th International Conference on Language Resources and Evaluation* (LREC 2004). Lisbon, Portugal.

[37] Patwardhan, S., Banerjee, S., and Pedersen, T. 2003. Using measures of semantic relatedness for word sense disambiguation. In *Proceedings of the 4th International Conference on Intelligent Text Processing and Computational Linguistics*. (CICLing'03, Mexico City). 241-257.

[38] Fellbaum, C. 1998. *WordNet: An electronic lexical database*. MIT Press, Cambridge.

[39] Achananuparp, P., Hu, X., and Yang, C. 2009. Addressing the variability of natural language expression in sentence similarity with semantic structure of the sentences. In *Proceedings of PAKDD 2009*. Bangkok. 548-555.

[40] Ho, C., Murad, M., Abdul Kadir, R. and Doraisamy, S.C. 2010. Word sense disambiguation-based sentence similarity. In *Proceedings of the 23rd International Conference on Computational Linguistic, COLING 2010*. Beijing. 418-426.

Augmented Reality for Tourism in Oman Using Free Open Source Software

Ahmed Al Maashri, Sara Al-Asadi, Muna Tageldin, Shurooq Al-Lawati, Ali Al Shidhani

***Abstract*— Tourism is an important and key economic driver in many parts of the world due to its high revenues. Tourism in Oman is induced by its spectacular beauty thanks to the geographic and environmental diversities that it contains. Therefore, Oman is looking forward to increase the contribution of tourism sector to the country's Gross Domestic Product. Because of these reasons, there are needs for creative tourism services to keep the flow of tourists coming to the country. Due to the diversity and disparity of tourist landmarks, there is a need for a portable and smart information dissemination service to provide rich information about the landmarks. The service has to be easily accessible, timely, and accurate. This paper presents the Portable Tour Guide, a free and open source solution targeting the tourism sector in Oman. The guide utilizes Augmented Reality technology to enhance the experience of the tourist using cloud computing services. Results of experiments on the implemented system show good accuracy and short response time.**

***Index Terms*— Augmented Reality, Oman Tourism, FOSS**

XXV. INTRODUCTION

ACCORDING to *Oman Observer* newspaper, more than two (2) million tourists from different nations have visited Oman in 2012. Compared to 2011, there is a significant increase by 48% [1]. Due to the importance of the tourism sector in Oman's economic diversification, it is forecasted that by the year 2024, tourism sector will contribute by 3.9% to the country's total Gross Domestic Product (GDP) [2].

However, one of the challenges that tourists face during their visit is the lack of tour guidance and proper information dissemination. For example, tourists may be interested in learning about the historical background of the landmark they just visited. Unless a professional tour guide is hired, it is difficult for the tourists to learn about the landmark and appreciate its significance to the country and to the world's heritage.

Therefore, a technology-compatible system is needed to provide the required information associated with landmarks. The objective of such a system is to enhance the tourists' experience by making the landmarks more appealing. This system should provide the tourists with a visual landmark-identification facility during their visit to Oman in real-time. More importantly, Augmented Reality (AR) technology is utilized to enhance the experience of the tourists by superimposing useful information on the image of the landmark when viewed on the tourists smart phone.

All authors are from the Department of Electrical & Computer Engineering, Sultan Qaboos University, P.O. 33, Al-Khodh, Muscat-123, Sultanate of Oman (phone: 968-24142539; fax: 968-24144336; e-mail: amaashari@squ.edu.om, alily@squ.edu.om)

We present Fig. 1 to demonstrate how AR is envisioned to guide the tourists in their visits. The figure shows a display screen of a smart phone, where the tourist has taken a picture of the landmark; in this case "Zawawi Mosque" in Muscat, Oman. Then, using a cloud computing service that is dedicated to this purpose, the smart phone will receive useful information that is superimposed on the original picture. As shown in the figure, a brief description is provided to the tourist.

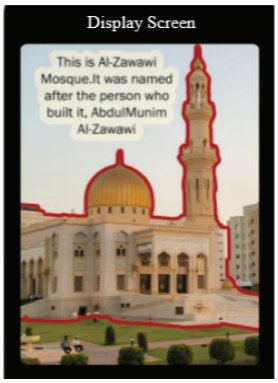


Fig. 1 A demonstration of the proposed system that uses AR technology to augment landmarks with useful information

The remainder of this paper is organized as follows: Section II presents related work, followed by a discussion of embedded vision systems and AR technology in Section III. Section IV details the system design of the Portable Tour Guide. Section V highlights system implementation and optimization, while Section VI discusses the obtained results. Finally, Section VII concludes the paper.

XXVI. RELATED WORK

Looking at the international tourism incomes in 2010 (a total of US $927 billion) and 2011 (a total of US $1,030 billion [3]), it can be stated that tourism is exceedingly vital to the world's economy. Many institutes have worked on developing various electronic products to serve the tourists and facilitate their tour trips. Basically, those products are divided into two main categories: General-Purpose Systems and Tourism-Dedicated Systems. These categories are described below.

A. Tourism-Dedicated Systems

Those devices are embedded systems designed for the purpose of tourism guidance only. These systems cannot be

used outside the tourism site. Some examples of these systems are listed below.

1) ARCHEOGUIDE

Augmented-Reality based Cultural Heritage On-site Guide or ARCHEOGUIDE is a device which aims to provide a tourist with information related to cultural heritage sites. The device consists of a transparent head-mounted display, a headphone, and a portable computing unit. The device is meant to be provided only at the targeted site. This portable computing unit has got a position tracking sensor, which interacts with some elements found in the site. After the position has been identified, the portable computing unit communicates with the site's server through a wireless local network and obtains the necessary information from the database about that specific location. Finally, the information is displayed to the visitor in a user-friendly way featured with 3D-visualization in real-time. In addition to displaying the superimposed image, ARCHEOGUIDE also speaks to the visitor through the headphones.

ARCHEOGUIDE was first tested in one of the well-known European cultural heritage sites in Olympia, Greece. Later on, several sites have implemented it. The disadvantages of this system are: (1) it requires high implementation effort, (2) expensive, (3) and serves specific sites only [4].

2) Audio Guidance Systems

Some historical sites in the UK use audio guidance systems in their tourism sites (e.g. castles). This technology is available for all visitors. When paying a nominal fee; each visitor receives a remote controller and headphones. While wandering around the castle, the tourist will find numbered tags placed in certain rooms, corridors, and monuments. The tourist uses the remote controller to choose the corresponding tag number and listens to the educational information about it. Other tour guidance systems may be huge screens placed at the entrance of a landmark. These devices show the map of the landmark and some other related information.

3) Prague Electronic Tour Guide

Prague electronic guide is a device that is designed for tourist areas in Prague city, Czech Republic. It is an audio device equipped with GPS module and buttons. The embedded GPS module provides the user with audio information regarding the tourist scenes that is close to the user. Theses audio information is updated while the tourist is walking around the city with the help of the GPS module. In addition, the device is able to save four locations to be re-visited by the tourist due to the presence of equipped navigational system [5].

B. General-Purpose Systems

This section presents multiple-use systems, where each system can be used for many applications including tourism guidance.

1) Google Glass

Google glass is a wearable device were the lenses are interactive. Using voice commands, google glass respond to user requests and show the result through lenses. It can provide variety of services to the user such as identifying Points of Interests (POI's) that are spotted by the tourist and giving directions [6].

2) Smartphone Applications

Tourism applications that are used mostly by tourists are classified based on the services they provide; namely, navigation, entertainment and information. Most of the existing navigational applications, which help the tourists find their way, are either AR-based or position-tracking based. An example of AR-based navigation applications is Layer app [7]. On the other hand, entertainment applications provide the tourist with a list of directions to entertainment places POI's (e.g. cinemas, theaters, festivals if any etc.), while information applications provide information about tourism in a specified country.

XXVII. Embedded Vision and Augmented Reality

Computer vision is a discipline that is concerned with emulating the human vision. This discipline combines both image processing techniques and intelligent algorithms in order to analyze scenes. With the abundance of computational resources in embedded systems, a new technology has emerged; namely, Embedded Vision Systems (EVS) [8]. This new technology allows small, but smart, devices to analyze scenes and make decisions accordingly.

There are a number of factors that made EVS a reality, including the availability of small inexpensive HD cameras, sufficient resources on embedded systems, and the breakthroughs made in the field of computer vision. On the other hand, researchers are investigating means to overcome challenges such as power consumption and real-time experience, to list a few.

Augmented Reality, AR, is one direct application of EVS that has gained a lot of attention recently. In essence, AR refers to augmenting real-world images with computer-generated multimedia such as graphics, text, audio, and video. AR has many applications in sectors such as education, industry, and tourism. AR relies heavily on computer vision techniques such as object recognition. For example, an embedded vision system may need to recognize a landmark, before it can supply useful information to the tourist.

Object recognition finds objects in the real world from an image of the world, using object models, which are known a priori [9]. Object recognition is usually a twofold process. First, features are extracted from the image. These features are representative of the object in the scene and can be used later to detect or recognize the same object when it appears in other images. These features has to be robust; in other words they need to be invariant to illumination, orientation, scale, and even affine. The second fold of the object recognition process is where these features are either matched to model object or classified into a predefined object category.

A. Speeded-Up Robust Features (SURF)[10]

SURF is an algorithm that extracts features from an image. These features, referred to as keypoints, are invariant

to translation, scale, orientation, and even affine. SURF is composed of several processing stages. The first stage is concerned with building a scale-space pyramid, which is controlled by the number and size of filters that are applied to the image. SURF gives the flexibility in choosing the layers (i.e. filter sizes) and octaves (i.e. groups of layers). Generally speaking, larger number of octaves and layers yield better accuracy, however, longer execution time. The authors of SURF specify 5 octaves, each of which hosts 4 different layers. This doesn't prevent the user from trading off accuracy for speed by reducing number of octaves and/or layers.

The remaining stages in the SURF algorithm are concerned with computing the orientation of each detected keypoint and describing these keypoints with a distinguishable value. Keypoints can be described either with 64 or 128 values. Although the 128-value mode (AKA Extended Mode) yields better accuracy, however, the execution time is slower when compared with the 64-value mode.

B. Brute Force Matching

Once keypoints are extracted, then we can attempt to find a match for the object at hand. The Brute Force matching is a commonly used technique that attempts to match each keypoint in the input image to all keypoints in the database. Also, it is worth mentioning that the standard metric for finding the distance between input keypoints and model keypoints is Sum of Squared Distance (SSD).

The following section discusses how these two computer vision algorithms are employed in order to support AR on the mobile devices of the tourists.

XXVIII. SYSTEM DESIGN OF PORTABLE TOUR GUIDE

This section describes the proposed Portable Tour Guide (PTG) system. We begin by outlining the engineering requirements that were identified by the authors to achieve the system's objectives. These requirements are:

1. System should superimpose text labels describing real-world images captured by the tourist.
2. System should permit the tourist to display more details about the captured image or show a nearby Point Of Interest (POI).
3. System should be accessible and easy to obtain.
4. System should not weigh more than 300g to make it comparable to the weight of mobile devices usually carried by tourists.

After identifying the requirements and examining existing systems, the authors compared several conceptual designs. Examples of such concepts are:

1. An embedded wearable device that contains a camera unit to capture photos, a processing unit to process data and a display unit (a screen) to show results. This device can be attached to the user's body and the display is held in the user's hand.
2. An embedded device with buttons, wireless access, a built-in GPS tracker, and headphones. When the device is turned on, the user's location will be processed. The device will receive audio information regarding the location and any nearby POI's. The user can use the buttons to turn the device ON/OFF, send location, repeat the audio information, and save audio.
3. A handheld device which points to a tagged POI to get information from it. The tag contains certain information about that POI. The device transmits electromagnetic waves that recognize the tag and takes necessary information about the POI from it.
4. An embedded device that has a camera, display, and a communication method to cloud computing service. The user takes pictures of POI and the images are sent via the communication method to the cloud for processing. The information related to that POI is sent to the embedded device for display (real-world images annotated with information).

Each of the proposed concepts has a set of advantages and disadvantages. Table 1 highlights the advantages and disadvantages of each design concept.

Table 1 Comparison between proposed design concepts

	Advantages	Disadvantages
1	- Allows the implementation of AR - Does not require wireless connection	- Hard to update for multiple users
2	- Doesn't require image processing - Portable	- High wireless bandwidth consumption
3	- Fast response - Allows the implementation of AR	- Impractical to attach tags everywhere - Limited availability
4	- Allows the implementation of AR - Doesn't require extra components other than the ones the user owns - Easy to update - Can be implemented in real-time	- Relatively high power consumption - High wireless bandwidth consumption

Based on the comparison shown in Table 1, it was found that concept number 4 is the most suitable. The most encouraging advantage of this concept is that its components (camera, communication means, etc…) are all incorporated in one embedded device. Hence, this reduces the overhead in the system because less interfacing between the components is required (unlike concept number 3).

Since, most AR existing products are smart phone applications, we have chosen smart phone to implement this system. The final product would be more appealing to the users since they are familiar with it. Moreover, choosing smartphone as the platform for implementing this system satisfies most of the system requirements. However, choosing the 4th concept brings forth two major challenges: (1) high power consumption and (2) limitations of communication bandwidth.

The proposed PTG system follows the client/server paradigm, as illustrated in Fig. 2.

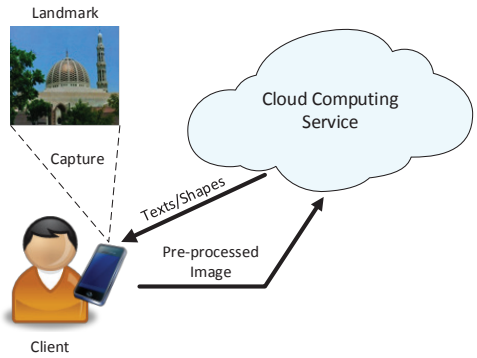


Fig. 2 PTG system adopts the client/server paradigm

A tourist (i.e. client) uses a smart portable device (e.g. smart phone) to capture a photo of the landmark that the tourist would like to inquire about. The captured image will be processed before being communicated to the cloud. In turn, the cloud will further process the received image, where extracted features are used to recognize the landmark. Once the landmark is recognized, the cloud will respond to the query of the tourist by sending descriptive text and shapes. Finally, the smart portable device will use the text and shapes to augment the captured photo as discussed earlier.

The following subsections detail the design of both client and server.

A. *Software Design of the Smart Device Application*

To facilitate the steps executed by the client, we propose the following process, which is depicted in Fig. 3.

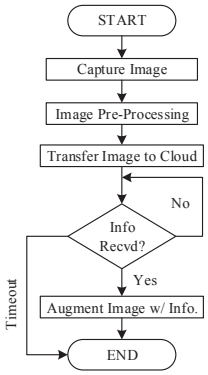


Fig. 3 Client side Process

The process starts by capturing an image frame of the landmark. Image sensors, found in smart portable devices, vary in the supported features such as frame resolution and bit representation of pixels. The software uses OpenCV library [11] to pre-process the images by resizing them to nominal frame size. In addition, the software converts the color space of the captured frame to grayscale. Pre-processing the frame serves two aspects. Firstly, making captured frames uniform in terms of resolution and color space, which helps in improving the accuracy of object recognition. Secondly, communicating captured frames in grayscale uses less bandwidth (*i.e. 8-bit grayscale vs. 32-bit RGB*), which results in reduced transmission delay. The software sends the processed image along with other information (e.g. image size) to the cloud.

At this point, the software awaits for the cloud to respond to the query. To avoid deadlocks, the software sets a timer such that if no response is received from the cloud, then the client can either try again or terminate the process if the cloud is unreachable. In case the cloud do respond to the query, then the software will augment the captured image with the information founded in the response. This process is then repeated for every captured image.

B. *System Design of the AR Cloud Computing*

The proposed Cloud Computing System is shown in Fig. 4. The system is composed of the following components:

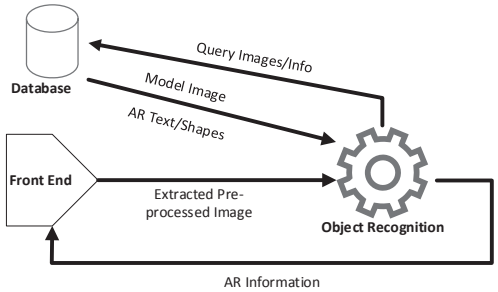


Fig. 4 The cloud computing service is composed of three components; front-end, Object recognition engine, and a database that hosts model images and landmarks information

1) *The front-end*

The system listens to incoming requests from clients. As soon as a proper request is received, the front-end will extract the image frame and other useful information from the payload. Then, this information is passed to the Object Recognition component.

2) *Object Recognition*

This component applies computer vision techniques on the received image frame. First, SURF algorithm is used to extract features from the image. Then, these extracted features are matched to pre-existing features. Once a match is found, then the landmark is identified. Then, the landmark is used to query the database for information relevant to that particular landmark. This information will be transferred to the client that made the request.

3) *Database*

The database is an essential component of the system. It hosts a repository of landmarks in Oman, historical information and interesting facts about each landmark, and other statistics (e.g. how many times this landmark was queried in the last month). In addition, the database implements a multi-level access system; allowing administrators to manage the content of the database.

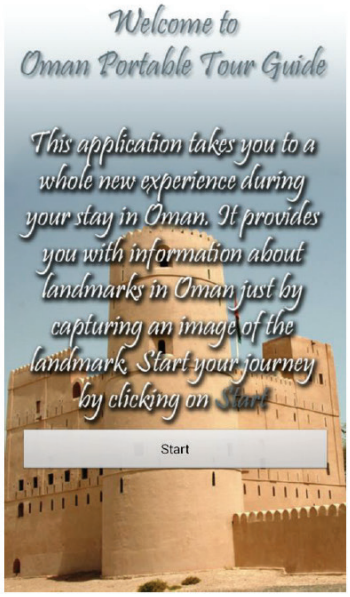
XXIX. System Implementation & Optimizations

The proposed PTG system for tourism was developed entirely with FOSS. The following subsections outline the details of the implementation.

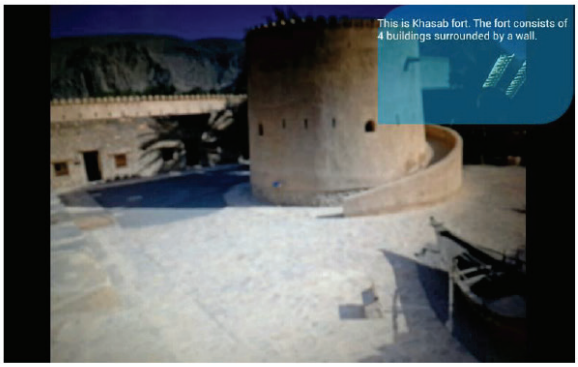
A. *Client Operating System and Application*

We choose Android OS [12] to operate the mobile device at the client side. This choice was made because Android is FOSS, widely used in embedded systems, and the development environment is readily available. Android SDK [13] was used to develop the application running at the client side. In addition, OpenCV for Android [14] was used for performing image pre-processing that was discussed in

Section XXVIII.A. Fig. 5 shows the user interface of the application.



(a) Welcome Screen



(b) Landmark enhanced with AR

Fig. 5 User interface of the mobile app, which is developed within Android platform

B. Server Operating System and Applications

The server is expected to be reliable and highly responsive. Linux CentOS 64-bit was used for the server platform. In addition, Eclipse IDE [15] was used for development, along with OpenCV C++ library for computer vision. In addition, XAMPP for Linux [16] was used since it combines the following package:

- **Apache server**: this webserver provides administrators a web-based interface to access the database for administrative purposes.
- **MySQL server**: this server facilitates database management.
- **PHP and Perl**: these two packages are mainly used for web development.

We anticipate that the landmarks repository will grow in the future. Hence, the database needs to be expandable. To facilitate that, a web-based interface is implemented for administrators; allowing them to easily upload new landmarks, edit existing ones, or delete a specific landmark.

C. System-level Optimizations

It is expected that as the number of tourists that use the proposed system increase that the server will be hammered with large number of client requests. In addition, as the size of the landmark repository increases, the matching stage in the Image Recognition component will dominate the execution time. This is because the server will attempt to match the features extracted from the query image to all features extracted from all model images. Therefore, it is crucial to design a scalable system to cope with such increase in demand.

In the last decade, we have witnessed an abundance of parallel processing capabilities, especially in the Many Integrated Core architecture (MIC) [17, 18] and Warehouse-Scale Computers (WSC) [19, 20]. The use of these parallel resources is the solution to develop a scalable system. Therefore, we have used the POSIX library to parallelize the workloads across the available processing cores. First, feature extraction of the query image was performed on a single core. Then, the extracted features are distributed across available cores. Each core has a subset of the model images' keypoints. Therefore, each core performs matching on a subset of the model landmarks, and hence reducing the total execution time for matching.

XXX. Discussion of Results

The proposed PTG system was prototyped and tested for execution time and accuracy.

A. Experiment Setup

For the client side, the developed application was installed on a Samsung Galaxy S4 smart phone [21]. The phone has 1.9GHz quad-core processor, 2GB RAM, and is operated by Android 4.2.2 "Jelly Bean" OS.

The server side was hosted on an Intel quad-core i7 machine clocked at 3.4GHz. The machine supports 64-bit architecture and contains 8GB RAM.

B. Design Space Exploration

This subsection provides an exploration of the design space. Since the machine vision algorithms dominate the execution time, we narrow down the dimensions of the design space to a number of octaves and layers implemented in the SURF algorithm.

Fig. 6 shows the relationship between number of SURF octaves and layers from one side, and execution time and accuracy, on the other side. The figure indicates that accuracy is improved by increasing number of layers and octaves. On the other hand, the execution time will also increase as a result of more computational load when increasing number octaves and layers. To balance between accuracy and execution time, we find that the optimum number of octaves and layers is 4 and 2, respectively.

C. Accuracy and Execution time

Next, we experiment with the system using the design configurations outlined in the previous section. We use an image dataset that is composed of 220 images for a total of 46 different landmarks. All images have a size of 800×600, since this size retains the main features of a landmark.

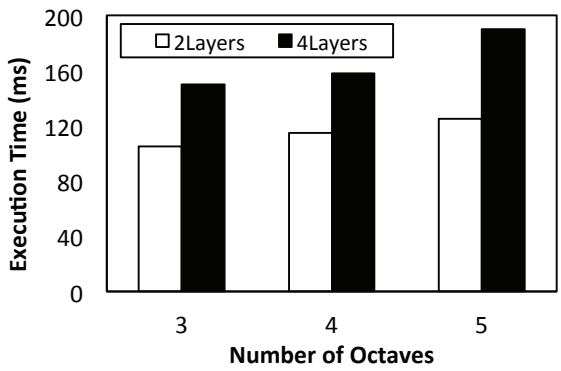


Fig. 6 Execution time of SURF algorithm using different configurations

Experiments reveal that on average, the total execution time is 936ms, where 62% of the execution time is dominated by the communication delay. On the other hand, the pre-processing that takes place in the mobile application (i.e. from the time an image is captured until the grayscale version of the image is packaged and ready for transfer) contributed by 78ms, while processing the image at the cloud took 278ms seconds.

We expect that the execution time at the client side will diminish since more powerful processors are being utilized in mobile devices and embedded systems. Similarly, the communication delay can be reduced by transferring compressed images that utilizes lesser bandwidth.

D. *Parallel Processing*

To reduce the execution time at the server side, we parallelize the matching process across the cores. Fig. 7 shows how increasing the number of utilized cores reduce the execution time. For example, when utilizing 8 cores, the overall execution time of object recognition is speeded up by 12.5%.

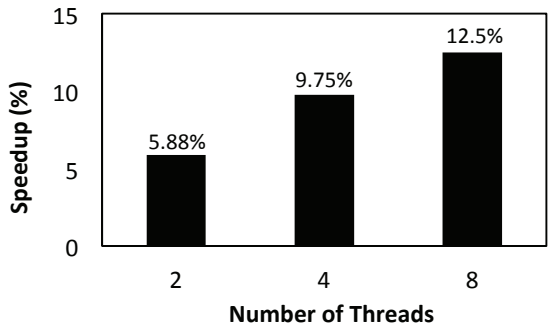


Fig. 7 Speedup in execution time when parallelizing the brute-force matching stage across multiple processing cores. Results are normalized to 1-core execution time.

XXXI. Conclusion & Future Work

This paper presented a proposed system that targets the Omani tourism sector. The proposed system was developed and implemented purely by FOSS software. The system was effectively deployed inside the campus of Sultan Qaboos University in May 2014.

For future work, we are planning on making further optimizations to improve the performance and accuracy of the landmark recognition. For instance, the system can make use of the GPS information to narrow down the number of points of interest (POI) to search for. Additionally, Object classification using Support Vector Machine (SVM) will be investigated.

References

[1] K. Yousuf. "Needed: More tourist guides," October, 2013; http://main.omanobserver.om/?p=24541.
[2] T. f. Oman. "Oman set for 41,000 jobs in tourism in 2014," http://www.timesofoman.com/News/33435/Article-Oman-set-to-lead-the-Middle-East-tourism-sector-in-2014.
[3] W. T. Organization. "UNWTO Tourism Highlights," https://s3-eu-west-1.amazonaws.com/storageapi/sites/all/files/docpdf/unwtohighlights12enlr_1.pdf.
[4] N. I. Vassilios Vlahakis, John Karigiannis, Manolis Tsotros, Michael Gounaris, Didier Stricker, Tim Gleue, Patrick Daehne, Luís Almeida, "Archeoguide: An Augmented Reality Guide for Archaeological Sites," *Computer Graphics in Art History and Archaeology*, 2002.
[5] E. T. Guide. "Prague Electronic Tour Guide :: about GPS audio-guide," http://electronictourguide.com/.
[6] Google. "Google Glass," http://www.google.com/glass/start/.
[7] A. R. Blog. "Augmented Reality," http://nipunsangral.blogspot.com/.
[8] E. V. Alliance. "What is Embedded Vision," http://www.embedded-vision.com/what-is-embedded-vision.
[9] R. K. Ramesh Jain, Brian G. Schunck, *Machine Vision*: McGraw-Hill, Inc., 1995.
[10] A. E. H. Bay, T. Tuytelaars, and L. Van Gool, "Speeded-Up Robust Features (SURF)," *Computer Vision and Image Understanding,* vol. 110, no. 3, pp. 346-359, 2008.
[11] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.
[12] Google. "Android," https://android.com/.
[13] Android. "Android SDK | Android Developers," https://developer.android.com/sdk/index.html?hl=i.
[14] OpenCV.Org. "Android | OpenCV," http://opencv.org/platforms/android.html.
[15] T. E. Foundation. "Eclipse," https://eclipse.org/.
[16] A. Friends. "XAMPP Installers and Downloads for Apache Friends," https://www.apachefriends.org.
[17] R. Wilson. "From Multicore to Many-Core: Architectures and Lessons," http://www.altera.com/technology/system-design/articles/2012/multicore-many-core.html.
[18] J. Jeffers. "Intel Many Integrated Core Architecture: An Overview and Programming Models," https://www.olcf.ornl.gov/wp-content/training/electronic-structure-2012/ORNL_Elec_Struct_WS_02062012.pdf.
[19] D. P. J. Hennessy, *Computer Architecture: A Quantitative Approach*, 4th ed.: Morgan Kaufmann, 2012.
[20] U. H. Luiz André Barroso, *The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines*, 2009.
[21] Samsung. "Samsung Galaxy S4," http://www.samsung.com/ae/consumer/mobile-phones/smart-phone/android/GT-I9500ZWAXSG?cid=ae_en_mob_s4_aon_ggl_ppc_dm_text_exact_20140918.

Open Source Software for Arabic Citation Engine: Issues and Challenges

Saleh Alzeheimi, Akram M. Zeki, Adamu I Abubakar

***Abstract*— Recently, there are various software for citation index such as Scopus, Google Scholar, Journal of Citation Report (JCR) and others. These software have some disadvantages such as not providing support to the literature which is written in Arabic and built as commercial and closed software. This study aims to explore Arabic initiatives in the development of OSS. An Arabic citation engine is an alternative to the existing software. It also aims to investigate the ability of Systematic Analysis for Arabic Citation Index (SAACI)**

***Index Terms*— Bibliometrics management software, Open source software, Citation Index, Systematic Analysis for Arabic Citation reference (SAACI).**

XXXII. INTRODUCTION

BIBLIOMETRICS is one of important scientific research tools for analyzing the characteristics of literature in specific subjects. This tool helps researchers and institutions to know relatives or separated subjects, knowledge sharing, the core of resources, impact factor of journals, high contribution among authors and other systematic analysis. According to Qasim [1], There has been increasing in number of journal requests to find the appropriate scientific tools to choose and evaluate the printed journals by librarians. In view of this importance, it has developed many systems for foreign mechanism which analyze intellectual foreign heritage, statistical indications and measurements of bibliometric. The most popular systems include, Scopus and Journal of Citation Report (JCR).

With the availability of many studies the mechanism of these systems is discussed, whether through the use of analysis of a particular intellectual heritage or through the evaluation of these systems, and the extent of the commitment for measurements and bibliometric, laws. But Arabic Studies (which means an analysis of Arab intellectual heritage) have not benefitted much from this system, as it does not support sources and citations written in Arabic language. This forces researchers and those who focus on bibliometric studies and analysis of characteristics of a particular intellectual heritage, to use the manual methods of analysis and application of bibliometric laws. With the existence of this gap, this study will attempt to explore Arabic initiatives in the development of OSS Arabic citation engine as an alternative to the existing software. The study will focus on answering two research questions:

Kulliyyah of Information and Communication Technology
International Islamic University Malaysia
Kuala Lumpur, Malaysia
zuhimi@gmail.com, {akramzeki, adamu}@iium.edu.my

- What are the current Arab initiatives in the development of open source software as the Arabic citation engine?
- What are the technical aspects and analysis for Systematic Analysis for Arabic Citation Index (SAACI)?

Hence, the answer to these questions will allow for the investigation of the ability and readiness of utilizing (SAACI).

XXXIII. RELATED WORKS

Among the studies, which deal with the characteristics of the Arab intellectual heritage, is the study by Timraz [2] who identified the characteristics of intellectual heritage that can be used by the Arabic researchers in the area of engineering in the Kingdom of Saudi Arabia. In the same year, Al Dausari [3] used in his study the bibliometric curriculum, entitled "An Information Communication among Arabia Researchers in Pure Sciences". In 1993, the study of Turkustani [4] use biometric law to explore the characteristic of intellectual heritage in the field of libraries and information In same year, Al-haddad [5] used a bibliometric analysis to study the citations in the Alam Alkotob magazine. In year 1994 Qandil [6] focused on analysing citations in the *Resalat Almaktabah* magazine. In year 2002 the study of "Electronic Publishing in Ten Years 1990-1999 Bibliometric study" by both Hamdi and Ghneim [7] aimed to identify the main features of an intellectual heritage. In 2003, Abdul Sattar [8] addressed in his study the electronic source for information in the area of library and information by analyzing citation references with Internet sources in Arabic periodicals. In 2003 Zayed [9] prepared a study on electronic source for distance citation references. She analysed theses, documentation and information in Cairo during the period of 1998-2003. In 2004, the study by Alumer [10] described characteristics of an intellectual heritage in the field of information technology through analysis of citations in Arab periodicals.

By tracing the tools used by the researchers in the above-mentioned mentioned studies, it is clear to the researcher, that these studies and others used manual methods to analyse Arabic intellectual heritage, where some of these studies used tables only to easily calculate averages and other simple small processes.

However, the study by Al-zeheimi [11] is the first Arabic study that developed the automated systems for open source, and was built on the web based for being most bibliometric measurements.

The second part of the literature is on Arabic studies that address applications for automated systems in the area of bibliometric studies. However, most of these studies are foreign studies except for the study by Kelow [12] and by Al-Najjar [13] who addressed the automated foreign

programs such as Scopus and JCR, but do not highlight the Arabic system, which was built and developed by Al-zeheimi [11,14], through his study of Omani intellectual heritage in the medical field. Al-zeheimi [14] has indicated this system and its characteristics in his paper presented at the conference of specialized libraries, in Arab Gulf in Kuwait in 2008.

To sum it up, it is observed that most of the previous studies used manual methods in Arabic intellectual heritage study, due to the weakness of potential automated systems that has been developed, such as Scopus and JCR in supporting Arabic publications and analysing their characteristics, as well as lack of Arabic intellectual heritage in electronic databases, and the rules of world information, compared to foreign intellectual heritage. Furthermore, there is no Arabic open source software built in web based, except the system for Systematic Analysis for Arabic Citation Index (SAACI) which was developed by Al-zeheimi [11,14], and used in his study on the Omani intellectual medical heritage.

Thus, the current study will address in detail the Systematic Analysis for Arabic Citation Index (SAACI) and its appropriateness in massive usage by specialists in this area, and the extent of its compatibility with the requirements of the Arab intellectual heritage.

XXXIV. METHODOLOGY

The study adopts an analytical method, to study the characteristics of the system for Arabs in the area of bibliometric studies, as well as addresses foreign systems, such as Scopus and JCR, in order to take advantage of the measurements, reports and statistics that are presented. Finally, the study highlightes an Arabic open source system (systematic Analysis for Arabic citation index - SAAIC).

A. *Systematic analytic tools*

The diffusion of systematic analysis tools or applications is increasing rapidly nowadays. The most popular of these tools are Scopus, Journal of Citation Report (JCR) and Google scholar.

1) *Journal Citation Report (JCR)*

Journal Citation Reports offers a systematic, objective means to critically evaluate the world's leading journals, with quantifiable, statistical information based on citation data. By compiling these articles' cited references, JCR helps to measure research influence and impact at the journal and category levels, and show the relationship between citing and cited journals. This is available in Science and Social Sciences editions (http://thomsonreuters.com/journal-citation-reports). JCR provides some bibliometric features such as Impact Factor, Immediacy Index, Total Cites, Total Articles, Cited Half-Life or Journal Title.

2) *Scopus*

As quoted from Wikipedia, cited from Scopus info; "*Scopus is a bibliographic database containing abstracts and citations for academic journal articles. It covers nearly 21,000 titles from over 5,000 publishers, of which 20,000 are peer-reviewed journals in the scientific, technical, and medical and social sciences (including arts and humanities)".* The sources informed that "*It is owned by Elsevier and is available online by subscription. Searches in Scopus incorporate searches of scientific web pages through Scirus, another Elsevier product as well as patent databases".*

B. *Arabic studies about systematic analysis tools:*

There are many Arabic studies that use systematic analysis tools (Scopus, JCR and Google Scholar) to analyze their literature. Researchers use these tools for English publication, but when they want to analyze any Arabic publications, they usually use manual tools and a few of them use excel or access software. The main reason for that is because of the unavailability of bibliometrics analysis tools that support Arabic publications. This causes some weaknesses in Arabic literature in different phases as follows:

- Weaknesses in quantitative studies that address the characteristics of Arabic publications compared to studies that deal with foreign once.
- Citations from books, dissertations and theses, patents and technical reports are poorly covered.
- Science subject is poorly covered.
- Weaknesses in quantitative studies that address the characteristics of the Arab intellectual output compared to studies that deal with foreign publications.
- Limitation in analysis. Most Arabic studies analyzed bibliographic data, but they ignore the reference citation.
- Weaknesses of comprehensive studies which implement the main bibliometric elements because researchers find it difficult to apply some bibliometric laws without software.
- Thus, this study aims to explore the existing tools which may provide a solution to overcome the weaknesses.

XXXV. ARABIC INITIATIVES IN DEVELOPMENT OF CITATION ENGINE/TOOLS FOR ARABIC PUBLICATIONS

Bibliometric / systematic / webmetrics tools like Scopus or Google Scholar aim to provide reports based on the three most commonly used laws in bibliometrics: Lotka's law of scientific productivity, Bradford's law of scatter and Zipf's law of word occurrence. Such reports like number of papers, citations, average number of citations per paper and per author and per year as well as h-indexs, g-index, and other metrics are provided by these tools. Most researchers who study the literature which has been written in English use a software to get bibliometric reports. This paper explores studies that analyze the literature which has been written in Arabic. It found that most studies analyzed the data manually to get biblimetric reports. For example, Temars [2] studied the citations in engineering in Saudi Arabia. He used the traditional tools to calculate the citations. In the library field, most researchers use excel sheet to get reports and graphs [1-10].

Based on these findings, the current study will analyse only an Arabic system analysis, the systematic analysis for Arabic citation index (SAACI), and after that it will be evaluating in terms of possibility of further development, with what the researchers required, and the characteristics of Arabic intellectual heritage, also with the latest laws on bibliometrics.

A. The idea of the software

The idea of designing a open source software based on MySQL and PHP came during the study of Al-zeheimi [11, 14] for Oman intellectual heritage, in the field of Medical Sciences, where the researcher found most Arabic studies using traditional methods for analysis.

The great potential of PHP language as its open source in report design is that it is possible to be flexible and have the potential for data and retrieval articles or citations, through the design of searching, or have the potential to search by more than one field at the same time. In view of the requirement of bibliometrics from automated expert systems, the focus on extract statistical indicators which has scientific indicators, and based on the bibliography requirement and citation references, the idea of designing a Systematic Analysis for Arabic Citation Index has resulted in SAACI [11, 14]

B. Advantages of the Software

The current study examines SAACI to identify the features and technical advantages. The study also analyzes the available elements in the system and come up with the following:

1- The open source software (OSS), allows source files modification and development, and allow developers to participate in the solution of the problems of the development program.
2- The ability to analyse the various types of materials such as periodicals, articles, books, University theses and manuscripts.
3- Ease of use; one of the advantages of the program is easy to move between user interface and screen, through the presence of hyperlinking elements such as main screen or articles, research or statistics and others.
4- Working in a network environment; the program works through the Internet, or through a network in the local area.
5- Multiple Entry where the design has several levels for accessing the program and to deal with it. There is the power for the director, which enables him to deal with all functions of the program. Also the Supervisor is entitled to know some of the statistics access reports and so on.
6- Linking the tables; the program has automated features that link tables, making it easy for the researcher to process data entry, for example after data entry for a particular article, the program provides hyperlink to enter citation reference contained in the article on the same screen.
7- Easy to search for articles: it is possible to use the program to search for articles in all fields, or to search for citations. The program provides method of linkage between fields, which can possibly specify a research process for accurate results.
8- Automatic construction of the lists of authors or publishers, through appropriate fields. This advantage saves effort, time and accuracy for the researcher in the process of entering data related to authors or publishers or topics. It is also easy, by clicking on the name or topic that appears automatically in the lists. Therefore, the program will show the users the process of entering author's citations and references of the article if the author had been named for the articles, in other words, the article had been entered earlier in the program, in the sense that the program is working to provide lists of the authors, topics and publishers.
9- Auto Update statistics; when entering data for a new article or an author or publisher or topic, the system will automatically update all the statistics that have been identified in the program. Upon completion of the process of entering the new article, for example; when entering a particular article under specific periodical year, the program automatically adds an additional number to the total articles entered, and adds an additional number to the total articles published based on the years, and adds an additional number to the total periodical articles and so on, which means that only the researcher can enter data or articles or citations at once, and get comprehensive statistical data.
10- Prepare the reports automatically. Reports that the program makes are divided automatically into two sections; the first section focuses on analysis of articles while the second section focuses on analysis of citations

1) C. Outputss of the Software

The program consists of five main elements, namely Articles, Citations, Reports, Management and Lists. Figure 1 shows the five main elements. Each main element contains sub-fields. For example, for Articles you can add full texts or just bibliographic data. Also, you can search in different fields. Reports are the other main element in this software which contains statistics about authors in articles, authors in citations, average of citations in each article, average number of authors in each article and citation. Reports also contain statistics about the characters of authors either in articles or citations. From this element, we can know the authors who have in their cited articles, the types of resources cited and other important reports which are authors who have been most cited in their articles.

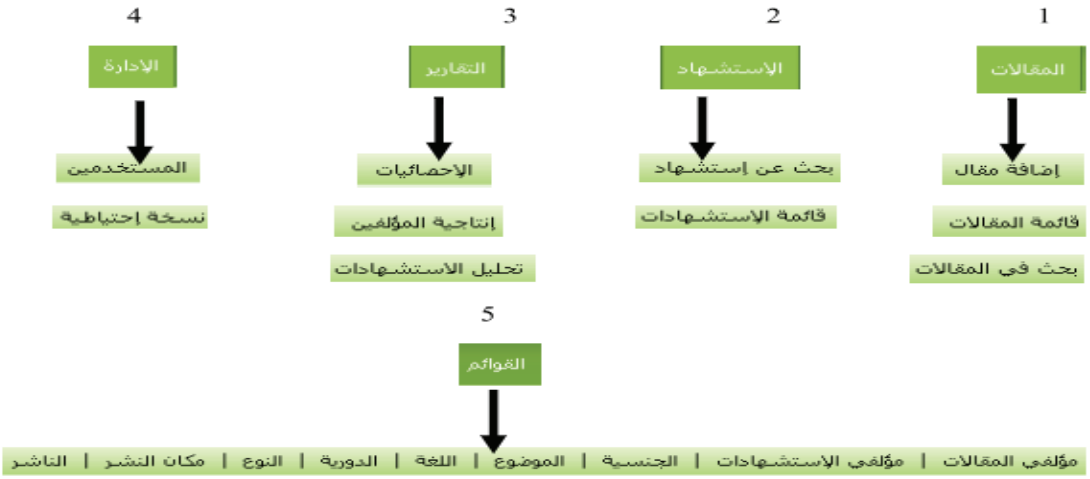


Fig. 1 The Five Main Elements and their Sub-fields.

1. *The main screen for outputs of the software*:
SAACI provides general statistics in the first screen for each user who has an account on this software. This part includes the total of number of studies that belong to the user, the total number of articles which he/she has inserted, over the total number of citation.

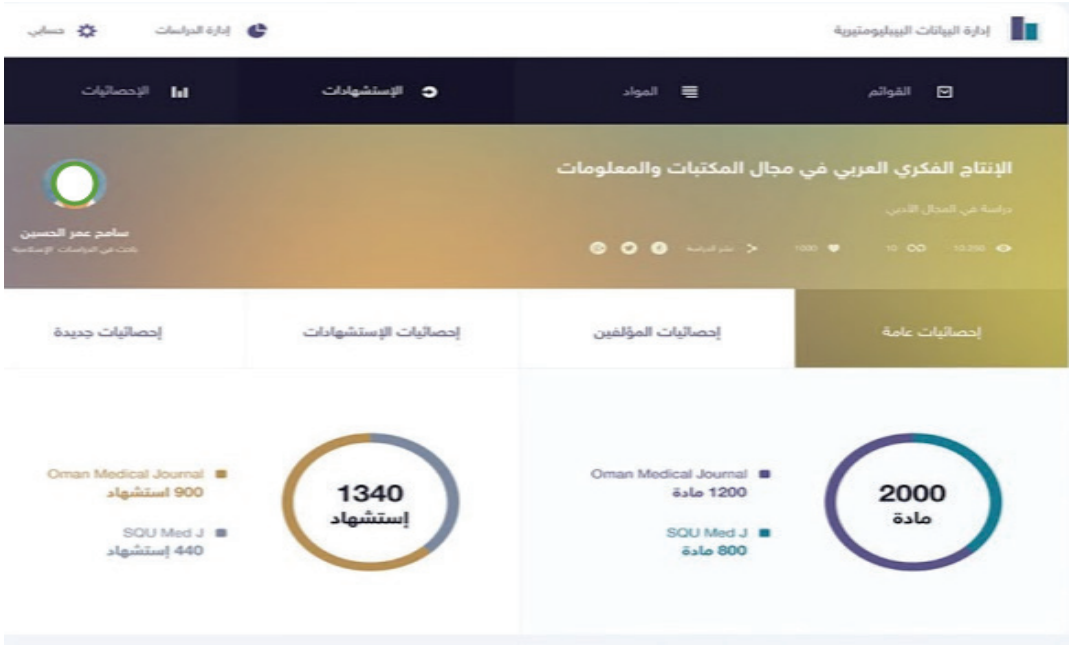


Fig. 2 General Statistics

2. *Authors characterized in articles*:
SAACI supports Arabic articles and its references and it helps the researcher to get the most important reports. SAACI divides reports into three parts: 1) Authors characterise in articles which include as follows:

- The most author published in the articles.
- The most author published in the references / citations.
- The most author published in articles / citations according to nationality.
- The rate of productivity of authors in the articles / citations
- Self-citation by authors.
- The rate of knowledge sharing among the authors in article / citations.

Fig. 3 The Characteristics of Authors in the Software

3. *Authors / journals characterized in the citations:*

This part in the SAACI provides the most popular reports based on bibliometric laws, which includes the following:

- The author who has the most citation in his articles.
- The author who has the most citation in his articles according to nationality.
- Self cited by journals
- The core journals
- Article distribution by subjects
- Article distribution by subjects in the citations
- Article distribution by publishing date
- Citation distribution by publishing date
- Citation distribution by sourcing types

The following figures show different reports which are available in SAACI.

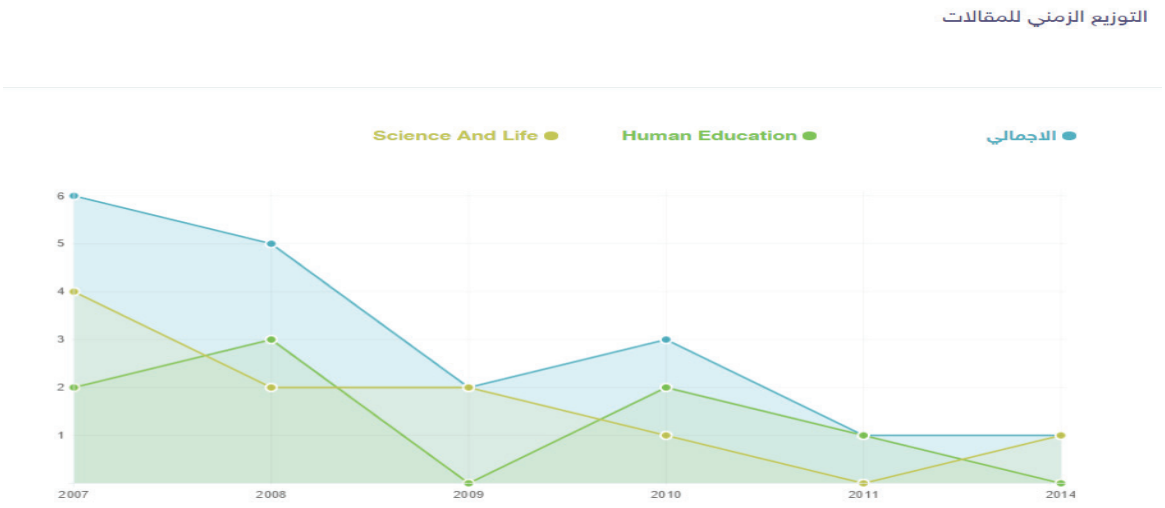


Fig. 4 Article Distribution by Publishing Date.

4. *The fourth part of the reports provided by SAACI is the following*:

- Bibliographic Coupling; a semantic similarity measure for documents that make use of citation relationships
- Cited Half-life
- Bradford Law of Scattering
- Obsolescence
- Impact Factor: the figure below shows the impact factors for Science and Life journal in 2009. This result can be based on the number of Arabic articles cited in it.

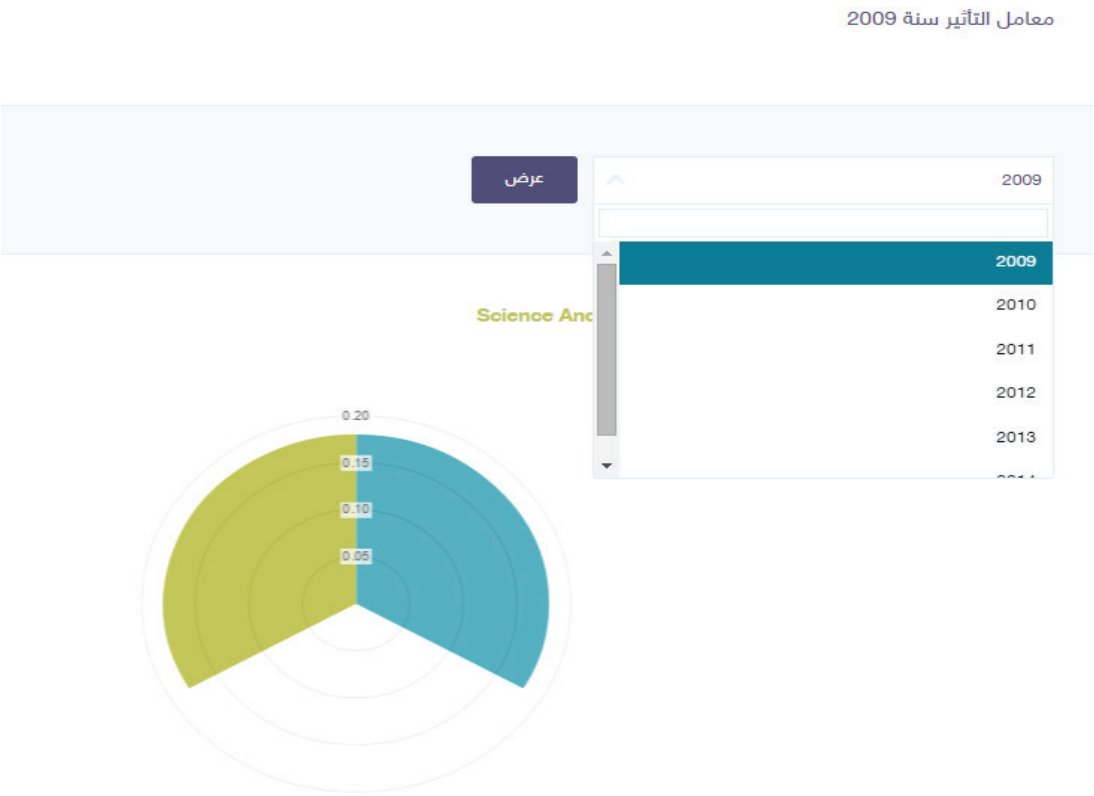


Fig 8 Impact Factor

XXXVI. Conclusion

This study identifies the issues and challenges in Arabic citation index. The results of the study indicate that the Systematic Analysis for Arabic Citation Index (SAACI) is the first Arabic open source software in the area of analyzing citation. The study demonstrates the advantages of the system which are easy to use and develop, due to it being an open source, and its suitability to all types of Arabic studies that focus on analysis of citation reference the program also analyzes most of the bibliometric measurements.

Acknowledgment

The authors wish to thank the Research Management Center and the Kulliyyah of Information and Communication Technology in the International Islamic University Malaysia for their supporting.

References

[1] Qasim, Heshmat (1984). Analysis of reference citations and the evolution of paper measurements. Cairo: strange library.

[2] Timraz, Ahmed Ali (1991). Intellectual output is used by Arab researchers in the fields of engineering properties in Saudi Asaudih.alm books, 12, (2) ,42 - 49.

[3] Al-Dosari, Fahd Mosfer (1991). Scientific contact at Arab researchers in pure science. Riyadh: King Fahd National Library

[4] Turkistani, Mohammed (1993). Intellectual production in the field of library and information: Bibliometrics study. Alam Al Kotob 14, (4) .462-463.

[5] Haddad, Faisal Abdullah (1993). Studies and reference citations in the magazine Alam Al Kotob: Bibliometric study. Alam Al Kotob 14, (4).

[6] Qandil, Yusuf (1994). Reference citations in the Journal of Resalat Almaktabah. Resalat Almaktabah , 29 (3) 23.

[7] Hamdi, Amal and Ghneim, Mohammed (2002). Electronic publication in ten years (1990-1999): Bibliometric study. Arabic studies in library and information science ,7, (2) 63-112.

[8] Khalifa, Abdul Sattar (2005). The use of an electronic source for information in the area of library and information through analysis of citation references with an internet sources in Arabic Periodicals articles. Arab 3000, 5, (3) 111 to 129.

[9] Zayed, Yusreya (2003). Electronic resources available in remote reference citations: analytical study of theses libraries, documentation and information department ethics Cairo: 1998-2003.

[10] Alumer, Haifa Ali (2004). Study an intellectual production properties in the field of information technology through the analysis of the reference citations in Arabic periodicals. (Master) Faculty of Social Sciences: University of Imam Muhammad bin Saud Islamic University, Riyadh.

[11] Alzeheimi, Saleh (2007). Intellectual production in Medical Sciences in Oman properties (1996-2006) Bibliometric study, (Master Thesis submitted to the Department of Library and Information Science. Sultan Qaboos University).

[12] Kelwo, Sabah (2009). Automated systems applications in the field of bibliometric studies and their impact on the creative management of libraries. Available on http://www.slideshare.net/SalehAlzuhimi/savedfiles?s_title=ss-14697719&user_login=SabahKallow

[13] Najjar, Mohamed (2006). Programs mechanism for the formulation of reference citations. Tenth conference for specialist libraries in Egypt. Cairo.

Alzeheimi, Saleh (2008). Bibliometric studies management system: a descriptive study. Conference of the Special Libraries Association branch of the Arab Gulf, Kuwait

System Quality Characteristics for Selected Mobile Platforms

Ahood Al-Darmaki [1] Nabeela Badursha [2] Ibtisam Al Shibli [3] Mohamed Sarrab [4]

***Abstract*— The number of mobile devices are increasing rapidly and manufacturers are competing to make the best device that satisfies the consumer's needs. With the huge number of devices available today, companies also developed special operating systems to differentiate their products from others. Some are open source and others are closed-source, each with special features and characteristics. This paper compares two of the most popular platforms iOS (as closed source platform) and Android (as open source platform) considering system quality characteristics including availability, reliability, performance, usability, and functionality. These characteristic are tailored to the criteria having been built from background study, standard for software quality and guidelines.**

Index Terms— *M*obile platforms, system quality characteristics, iOS, Android.

XXXVII. INTRODUCTION

SOME software has source code that cannot be modified but the person, team, or organization who created it maintains exclusive control over it. This kind of software is frequently called "proprietary software" or "closed source" software, because its source code is property of its original authors, who are the only ones legally allowed to copy or modify it. Internet Explorer, Adobe Reader, Adobe Photoshop, VMS, Microsoft Office etc. are examples of proprietary software [1]. In order to use proprietary software, computer users must agree (usually by signing a license displayed the first time they run this software) that they will not do anything with the software that the software's authors have not expressly permitted. Open source software has different concept. Its authors made the source code available to others who would like to view that code, copy, learn, modify, or share it. Anypic, ifixit, Ethita, Vim, istrobe etc are examples of open source software for iOS and BlueGPS, Kontalk, Moss, VLC etc are examples of open source software for Android. As they do with proprietary software, users must accept the terms of a license when they use open source software but the legal terms of open source licenses differ dramatically from those of proprietary licenses.

[1,2,3]Communication and Information Research center, Sultan Qaboos University, P.O. 17, Al-Khodh, Muscat-123, Sultanate of Oman (phone: 968-24143697; fax: 968-24144336; e-mail: ahood@squ.edu.om, anabeelapb@squ.edu.om, Ibtisamq@squ.edu.om

[4]Communication and Information Research center, Sultan Qaboos University, P.O. 17, Al-Khodh, Muscat-123, Sultanate of Oman (phone: 968-24143698; fax: 968-24144336; e-mail: sarrab@squ.edu.om

Open source software licenses promote collaboration and sharing because they allow other people to make modifications to source code and incorporate those changes into their own projects. Some open source licenses ensure that anyone who alters and then shares a program with others must also share that program's source code without charging a licensing fee for it. In other words, mobile application developers can access, view, and modify open source software whenever they like as long as they let others do the same when they share their work [2].

XXXVIII. MOBILE PLATFORMS

The mobile platform also refers to mobile operating system, is an operating system that is specifically designed to run on mobile devices such as mobile phones, smart phones, PDAs, tablet computers and other handheld devices. Mobile operating system is the software platform on top of which other programs, called mobile application programs, can run on mobile devices [3][4]. It is responsible of managing between hardware and software, performs basics tasks, management of memory, and ensures that different applications are running at the same time without any interference. The types of mobile platforms (based on existing operating system used by computer) [5]:

a. Real-Time operating system,
b. Single user, single tasking operation system,
c. Single user, multi-tasking operating system, and
d. Multi-user operating system.

XXXIX. QUALITY CHARACTERISTICS

System quality implies the desirable characteristics of the mobile platform product. The selected criteria differ from the empirical study of open source software selection for adoption based on software quality characteristics [6].

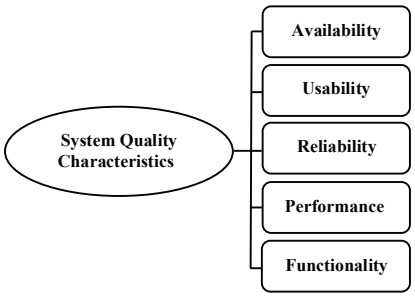


Fig. 1 The used quality characteristics to select Mobile Platforms

Also it differs from the selection criteria of open source software: first stage for adoption [7] which involves system

quality, information quality and service quality. Our approach considers the system quality characteristics that include availability, usability, reliability, performance and functionality. The selected criteria may contribute to the success of mobile platform development and adoption among potential developers and users [8][9].

XL. THE STUDY

The compared platforms are iOS and Android, two of the most popular mobile operating systems. Android mobile operating system is based on the Linux kernel and Google's open and free software stack that includes an operating system, middleware and also key applications for use on mobile devices, including smart phones. Updates for the open source Android mobile operating system have been developed under "dessert-inspired" codenames (Cupcake, Donut, Eclair, Gingerbread, Honeycomb, Ice Cream Sandwich, Jelly Bean, KitKat, Lollipop) with each new version arriving in alphabetical order with new enhancements and improvements.

Android technology is based on Java. In order to create applications in Android, developer has to know software development kit (SDK). The SDK can be downloaded freely from the Internet and can work on multiple operating system, that's why many developer prefer development using SDK. Also, they can sell their creations for many consumers [5]. Android system is programmed in C, C++ and JAVA languages [10]. The Android developers use the open source code as a foundation for community driven projects that add new features for advanced users. Android's API allows exploring the device's capabilities to give the application chance to adapt accordingly.

Apple's iPhone OS was originally developed for use on its iPhone devices. Now, the mobile operating system is referred to as iOS and is supported on a number of Apple devices including the iPhone, iPad, iPad 2 and iPod Touch. The iOS mobile operating system is available only on Apple's own manufactured devices as the company does not license the OS for third-party hardware. Apple iOS is derived from Apple's Mac OS X operating system. iOS system is programmed in C, C++ , Objective C and Swift languages.

The multi-touch interface, 3D components and flawless usability makes it widely used among Smartphone user interfaces. iOS limits access to some websites as it does not support Java or Flash currently. Apple iOS developers were allowed to develop applications for iPhone and iPod Touch using SDK on March 6, 2008. Apple also enabled application test on " iPhone simulator". Downloading applications into devices are allowed only after paying iPhone Developer Program fee [11].

A. *Availability*

Availability of the mobile platform means the availability of latest version and upgrades for the latest released devices. The availability of mobile platform in different languages is very important. Any upgrade in the software version or addition of new features should be made available to the mobile platforms at the expected time.

Table 1. Mobile Platform availability

Mobile Platform	Availability
Android	Google provides major upgrades, incremental in nature, to Android every six to nine months. Compared to iOS, Android updates are typically slow to reach actual devices [12]. Android is available in about 57 languages [13].
iOS	Apple provides update to the iOS almost once a year. 90 % of devices are using the latest version [14]. Most iPhones are allowed to make upgrades from one generation to another [15]. iOS supports around 36 languages [14].

B. *Usability*

Usability in mobile platforms refers to the ease to master the platform and get used to its features. It also refers to how easily the platform operates and accessed without a third party software. Usability of mobile platforms has increased greatly in recent years allowing users to perform more tasks in a mobile context.

Table 2. Mobile Platform Usability

Mobile Platform	Usability
Android	Android devices had front-facing cameras first but relied on third party apps to handle the video call, even over 3G, but with decidedly mixed results. This complexity makes it less useable. [16]
iOS	iOS consistency, attention to interface and experience detail, and level of fit and finish make it just as inarguably more usable [17]. iOS leaves an impressive 70% of its users very satisfied with the experience of using it.[18]

The ISO standard outlines the usability criteria in terms of three attributes; effectiveness, efficiency and satisfaction.

- Effectiveness: Accuracy and completeness with which users achieve specified goals;
- Efficiency: Resources expended in relation to the accuracy and completeness;
- Satisfaction: Positive attitudes towards the use of the product.

Issues such as the small screen size, poor connectivity and limited input modalities have an effect on the usability of mobile platforms. The factors such as User (Person who interacts with the product), Goal (Intended outcome) and Context of use (Users, tasks and the physical and social environments in which a product is used) are considered while evaluating usability criteria for mobile platforms [6].

C. *Reliability*

Mobile platform reliability refers to how mature the platform is and how many years it has been up in the market. It also refers to how much stable and secure the

platform is. Reliability also refers to the support available in case of issues and tolerant. Reliability is an important attribute of software quality. Software Reliability deals with the probability that software will not cause the failure of a system for a specified time under a specified condition which reflects the design perfection. The tools such as RGA 7 from ReliaSoft and SMERFs (Statistical Modeling and Estimation of Reliability Functions for Software) are used for testing the reliability of mobile platforms [7].

Table 3. Mobile Platform Reliability

Mobile Platform	Reliability
Android	Android is about 6 years old [19]. It lacks a complete stable version due to many released firmware numbers. According to a report from FixYa, android is 187 % reliable than iOS [15].
iOS	iOS is 7 years old [20]. Apple's tech support is quick, friendly and helpful. One can get help through phone call, live chat, or simply the support webpage [21]. iOS is more secure from malware applications and most applications are designed to be compatible to any iOS version, which makes it more reliable [15].

The reliability of the application depends on the skills of the developer and the tester. The five key challenges faced while assessing and improving the reliability of mobile platform are:

- Impact of context on reliability: There is a wide spectrum of contextual properties that may influence the reliability of a software system: changes in the hardware platform (e.g., depleted battery, memory usage), fluctuations in the network (e.g., network drop outs, bandwidth variations), unanticipated usage, timing of operations, and so on.
- Impact of dynamism on reliability: Allocation of software components to OS processes directly impact the reliability of a software system. Therefore, in the mobile computing domain, where architectural reconfiguration occurs frequently, the ability to assess its impact on reliability and determine the most reliable configuration is critical.
- Difficulty of predicting reliability: The ability to determine an ideal reliable configuration for a software system depends on the ability to accurately estimate the reliability of the software system in its future operation.
- Quality reliability prediction: The configuration is considered preferable if the components are extremely reliable and efficient where as if the components are highly unreliable it is more resilient to failure.
- Scalable online analysis: The scalability of analysis is extremely important for the platform that is performed online at runtime [22].

D. Functionality

Functionality is a feature that depends on usability and it refers to achieving the user's expected requirements. It refers to the features and the functions the platform can do to satisfy the user's needs.

Table 4. Mobile Platform Functionality

Mobile Platform	Functionality
Android	Users can do more in android than they could in any other OS, which makes it functional [16]. Google keeps adding new functions to its new android versions like printing on the go, built-in sensors, low power location monitoring, easy home screen switching, built-in infrared blaster support and a redesigned downloads app [23].
iOS	The newer upgrade of the iOS named iOS 8 has many functions cannot be found in any other mobile OS, like connecting all kinds of iDevices, sending any kind of messages from any device and answering calls from any device. [24].

E. Performance

Performance is the most important differentiator when it comes to mobile platform adoption. Performance is mostly about speed; how easy it is to install, configure and operate within a short time. The design of the user interface (UI), is an essential part for the performance of mobile platform. The design of user interfaces is necessary to be compatible with the demands of users [25].

Table 5. Mobile Platform Performance

Mobile Platform	Performance
Android	The overall performance is excellent except for the compatibility issues [26].
iOS	Everything from software updates to automated data backups can be managed effectively using the iTunes interface, which gives it a fast overall performance without the need to any tweaking [16].

XLI. Discussion

The above five tables showed the two compared platforms with respect to the five features; availability, usability, reliability, functionality, and performance respectively. The availability in iOS is considered better than android, because each new iOS version is backward compatible with the older devices. In the other hand, a new android version is not necessarily compatible with the older devices. In fact, android updates are mostly specified for only specific devices manufacturers. However, looking at the languages supported by the two platforms, it is clear that android beats iOS by a big deal by supporting 21 more languages than iOS. Usability and functionality are two related features. More usable means less complexity which comes from the functions. iOS is considered more usable. As it has a very simple interface and most of the management tasks that are not needed by the user are hidden behind the scene. Unlike android, its interface is more complex and users can get messages related to memory

management and other things. However, android is more complex and can do more functions than iOS. Looking at the reliability of the two platforms, iOS wins the battle. iOS is more stable and secure which makes a huge percentage of users move to the newer versions of it. This helps apple to concentrate on the newer version and finding issues and fixing them. In the other hand, android does not have a stable version since it is being developed by many companies. So, different devices have different versions of android depending on the manufacturer. The performance of the two platforms can be considered the same. Both have good performance and good techniques to maintain higher performance for a better user experience.

XLII. Conclusion

Although mobile platforms are relatively similar with respect to their hardware capabilities, they differ greatly in their native application development models. Android is a mobile operating system running on the Linux kernel that uses Java as the development language. iOS use Cocoa Touch as user interface that developed using Objective C language. The user interface of Android simplifies internationalization and allows depicting the user interface on different screen resolutions. Cocoa Touch offers a variety of UI elements. Unlike Android, Cocoa Touch offers no layout manager so the UI elements have to be positioned in terms of absolute coordinates on the screen at design time.

Android applications can more easily adapt to different devices that makes them cross compiled to other platforms. Apple iOS is fairly hostile when it comes to other platforms and requires developers and companies to submit an application for a service they want to add to its OS. Android has the largest installed base worldwide on smart phones and iOS stands second. Android platform offers high performance due to its multitasking capabilities and are fast and easy to access. iOS is less virus, fast boot and software are compatible with the operating system. The transfer of multimedia files is limited in iOS.

Acknowledgment

This article is based upon research work funded by The Research Council (TRC) of the Sultanate of Oman, under Grant No: ORG/SQU/ICT/13/006, (www.trc.gov.om).

References

[1] Sarrab M, Mobile Learning (M-learning) Concepts, Characteristics, Methods, Components. Platforms and Frameworks, Nova Science Publishers , Newyork, USA, 2014. ISBN 978-1-63463-342-0

[2] Sarrab M, M. Elbasir and L. Elgamel, An Overview of the Technical Challenges of Migration to Open Source Software. Free and Open Source Software Conference (FOSSC-13), Muscat Oman. pp 39-42. February 18th- 19th, 2013.

[3] Vangie B, Mobile Operating Systems (Mobile OS) Explained, webopedia. August 10,2011 http://www.webopedia.com/DidYouKnow/Hardware_Software/mobile-operating-systems-mobile-os-explained.html

[4] Alzahrani A, Alalwan N, Sarrab M. Mobile Cloud Computing: Advantage, Disadvantage and Open Challenge, Proceedings of the 7th Euro American on Telematics and Information Systems, (EATIS 2014); 2014 Article No. 20 ACM New York, NY, USA.

[5] Gaurav Jindal, Mayank Jain, "A Comparative Study of Mobile Phone's Operating Systems", November2012, International Journal of Computer Applications & Information Technology Vol. I, Issue III

[6] Rachel Harrison, Derek Flood and David Duce, "Usability of mobile applications: literature review and rationale for a new usability model", 2013, Harrison et al. Journal of Interaction Science.

[7] Sam Malek , Roshanak Roshandel, David Kilgore , Ibrahim Elhag, Improving the Reliability of Mobile Software Systems through Continuous Analysis and Proactive Reconfiguration.

[8] Sarrab M, Rehman O. Empirical Study of Open Source Software Selection for Adoption Based on Software Quality Characteristics. Advances in Engineering Software, Elsevier. Vol. 69, pp. 1-11, March 2014.

[9] Sarrab M, Rehman O, Selection Criteria of Open Source Software: First Stage for Adoption, International Journal of Information Processing and Management (IJIPM), Vol. 4, No. 4, pp. 51-58, 2013.

[10] Kristen Dietiker, "Managing iOS Mobile Devices", November 2011, ACM.

[11] Mark H. Goadrich, Michael P. Rogers, "Smart Smartphone Development: iOS versus Android", 2011, 978-1-4503-0500.

[12] Mike Isacc, A deep-dive tour of Ice Cream Sandwich with Android's chief engineer, Oct 21 2011, http://arstechnica.com/gadgets/2011/10/a-deep-dive-tour-of-ice-cream-sandwich-with-androids-chief-engineer/

[13] Apps4android. Languages and Locales Supported by Android. apps4android.org. April 15th, 2012, http://www.apps4android.org/?p=3695

[14] Apple Inc, App Store Distribution,2014, https://developer.apple.com/support/appstore/

[15] Ryan Inoyori "iPhone Tops Reliability Survey, Better than Android Phones" February 27, 2013, http://au.ibtimes.com/articles/439836/20130227/iphone-tops-reliability-survey-android-phones.htm#.U-MN19bHiRu

[16] Bryan Wilde, "iOS vs Android vs Windows: The Battle of the Mobile OS", November 22, 2013, http://www.udemy.com/blog/ios-vs-android-vs-windows/

[17] Rene Ritchie, "iPhone, Android, and the difference between usability and functionality [Sticky]", Dec 22, 2010, http://www.imore.com/iphone-android-difference-usability-functionality.

[18] Nick T, "Apple's iOS tops the customer satisfaction charts; Android holds third place", 21 Jul 2011, http://www.phonearena.com/news/Apples-iOS-tops-the-customer-satisfaction-charts-Android-holds-third-place_id20512

[19] Dan Morrill, "Announcing the Android 1.0 SDK, release 1", 23 September 2008, http://android-developers.blogspot.in/2008/09/announcing-android-10-sdk-release-1.html

[20] Mathew Honan, "Apple unveils iPhone", Jan 9, 2007, http://www.macworld.com/article/1054769/iphone.html

[21] Michael Andronico, "Apple Tech Support: 2014 Rating", Sep 30, 2014, http://blog.laptopmag.com/apple-tech-support-2013

[22] Sonia Meskini, Ali Bou Nassif, Luiz Fernando Capretz, Reliability Models Applied to Mobile Applications, 2013, Seventh International Conference on Software Security and Reliability Companion.

[23] Azzief Khaliq, "Android 4.4 KitKat: 10 New Features You Should Know",http://www.hongkiat.com/blog/android-kitkat-features/

[24] Apple Inc, "iPhone, iPad, and Mac. Connected like never before.", 2014 http://www.apple.com/ios/ios8/continuity/#gallery-handoff-section-3

[25] Farhad Soleimanian Gharehchopogh, Farzaneh Abbaspour, Maryam Tanabi, Isa Maleki, "Review and Evaluation of Performance Measures in the Mobile Operating Systems", March-2013, International Journal of Scientific and Engineering Research Volume 4, Issue3 ISSN 2229-5518.

[26] Andrew Williams, "Google Nexus 5 - Android 4.4, Apps and Performance", 5 Nov 2014, http://www.trustedreviews.com/google-nexus-5_Mobile-Phone_review_android-4-4-apps-and-performance_Page-3

Shaima M. Shakeel[1], Osama M. H. Rehman[1], Mohamed Ould-Khaoua[2], Hadj Bourdoucen[1]

Open Source Software Support for Field Experiments of Vehicular Ad Hoc Networks

***Abstract*—The ever increasing need of technology-users to have greater levels of stability, control and security over the hardware has paved a long way in the field of free and open-source software (FOSS). The FOSS-based powerful tools have a plethora of uses in the field of communication. Vehicular ad hoc networks (VANETs) is a part of the intelligent transportation system(ITS), which is a rapidly growing area of research since the past decade.VANETs aim to provide better traffic management, such as the reduction ofroad accidents and traffic congestion. This paper sheds some light on the usage of FOSS tools to conduct field tests in VANETs. Various open source tools such as Linux operating system, C programming language, AWK scripting and Wireshark network packet analyzer have been deployed to study and evaluate VANETs. This paper also presents the benefits and drawbacks faced while using the FOSS tools to carry out the field experiments on VANETs.**

***Index Terms*— C++, Free and open source software (FOSS), Linux, VANETs, Wireshark.**

XLIII. INTRODUCTION

Vehicular Ad Hoc NETworks (VANETs) is considered as one of the state-of-the-art field of research [1]. Since the mankind has begun to witness the development of roads and means of transportation to ease and comfort the traveling experience, it has been inevitable to hear of negative effects of it as well. In other words, these developments made in transportations, in order to ease our lives, goes hand in hand with being a great source of physical, financial and psychological risks.

Recently, the field of vehicular communication and networking had gained a momentum for development [2]. With the recent technological advances in various fields, a new trend of communication referred to as the ad-hoc network has come into existence.This is a special type of decentralized wireless networks, where each communicating device in the network (referred to as a node) has the property of arranging itself and acting as an end-node and as a router in a very dynamic fashion, as and when required. One of the types of ad-hoc networks is the Mobile Ad-Hoc NETworks (MANETs), where nodes are mobile and free to move independently and hence the links between these nodes also alters frequently [3,4].

[1]Communication and Information Research center,Department of Electrical & Computer Engineering, Sultan Qaboos University, P.O. 17, Al-Khodh, Muscat-123, Sultanate of Oman (phone: 968-24143696; fax: 968-24144336; e-mail: m100170@student.squ.edu.om,p091608@student.squ.edu.om, hadj@squ.edu.om)

[2]Department of Electrical & Computer Engineering, College of Engineering, Sultan Qaboos University, P.O. 33, Al-Khodh, Muscat-123, Sultanate of Oman (phone: 968-24142537; fax: 968-24413454; e-mail: mok@squ.edu.om)

A subcategory of MANETs which has recently been a hot topic for research is the area where the MANETs span the vehicular movement, such as road vehicles, automobiles, trains etc. This important subcategory is classified in general as VANETs.

VANETs are used to facilitate inter-vehicle communication with the intention of improving the road safety, efficiency and comfort of the road journey in day-to-day life. This type of communication can work without the need for expensive infrastructure. VANETs rely on direct communication between vehicles to satisfy the communication needs of a large class of applications (e.g., collision avoidance, passing assistance and platooning). VANETs systems can be supplemented or, in some situations, replaced by roadside infrastructure, allowing for Internet access and several other applications [5].

In VANETs, each vehicle is considered as a node which will act as a router to transmit or receive information from the other nodes on the road. An on-board unit (OBU) within the car will help the car to receive warnings for the driver and act accordingly to prevent accidents. Each car can have a range of communication (100m to 1000m) depending on the power or transmission, to send or receive information [6]. In such a situation, the other cars that join in to form a new network and act as intermediate nodes to transmit or receive the signal, and hence, a mobile network is created.In order to study VANETs under different scenarios, two methods have been most widely used:

1) Vehicular ad hoc networks *Simulation*: Various VANET simulation softwares are free and open source such as the traNS, NCTUns, GrooveNet, MobiREAL are used. In addition, the network simulators which are most commonly used are ns2, ns-3, SNS, GloMoSim and GTNetS[7].
2) *Field testing/ prototyping*: where the actual deployment of the technologies are put to test in real life scenarios on the vehicles on road use the required equipment for communicating between on-road vehicles

Various research projects have given solutions for fulfilling the technological requirements in VANETs while using simulations software. However, current focus is shifting from simulated-based research to practical implementation in the form of on-field operational tests [8,9]. The authors in [10,11] have carried out various field tests to evaluate the vehicle-to-vehicle and vehicle-to-infrastructure applications in the real world scenario to prove that VANETs can indeed enhance road safety, traffic efficiency and travel comfort. Moreover, a number of researchers have designed an enhanced emergency warning system [12] and tested it in order to evaluate its performance.

This paper aims to demonstrate the effectiveness of expending the FOSS tools in performing field testing in VANETS and discusses some of the short comings faced in the process of deploying these open source tools.

The rest of the paper is organized as follows. Section II discusses the arrangement of the testing environment and provides a thorough description of the FOSS tools used in the implementation of field experiments. Section III briefly discusses and comments on the results obtained from testing in the static nodes scenario. Section IV highlights some of the shortcomings faced by the deployment of the FOSSC tools in these experiments. Finally, Section V mentions the

challenges faced during this study and the future works to be conducted in this study.

XLIV. FOSS TOOLS IN VANETS TEST ENVIRONMENT

A. FOSS Tools

1) Linux operating System

The popularity of this operating system can be justified by the fact that it is capable of supporting a host of hardware platforms including audio & video devices, network cards, GPS, laptops and notebooks, motherboards, cameras and antennas to mention a few [13]. It also provides an efficient platform to run open source network simulation softwares such as NS2 [14]. Another significant feature of Linux based operating systems is the ability to create and support efficient links in wireless Ad Hoc networks with ease. And hence, all these features make the operating system a suitable FOSS tool for performing tests in VANETs.

The field tests conducted in this study has been implemented using the latest versions Ubuntu, namely: 12.04 and 14.04. Ubuntu is free and open source software which works on a Linux based kernel.

The equipment and the devices used for the tests are tabulated in Table 1.

Table 1: List of Devices used to Field tests

Hardware Device	Device Brand	Device Model
2.4GHz 8dBi Indoor Desktop Omni-directional Antenna	TP-Link	TL-ANT2408C
USB Wireless WiFi Network Adapter	Alfa	AWUS036H
Mouse GPS	Globalsat	BU-353S4
Data Processing Units (Laptop)	Toshiba	Satellite C660-24E

2) C Language & AWK scripting language

The C language is the base of all the coding involved in the kernel and auxiliary softwares developed in Linux based operating system. It was essentially created for the purpose of designing system based and embedded software programming. It is the most commonly used languages for designing and building open source software tools [15].

AWK is yet another built in feature of Linux operating systems, which is a data extraction tool written for manipulating streams of textual data [16]. It is a very flexible and efficient tool when it comes to processing and extracting specific set of data from a large chunk of pipelined data or a large data file.

For the field tests on VANETS, socket programming was used for inter process communication protocol between sender and receiver nodes in the network. There are two types of sockets that are commonly used: TCP (stream) sockets and UDP (Datagram) sockets. Another type of socket is the RAW socket, which allows direct access to lower-layer protocols, such as Internet Protocol (IPv4 or IPv6) and Internet Control Message Protocol (ICMP or ICMP6). [17].

C language has been used for creating the socket programming in the field experimental tests conducted in this work. It has been compiled using the GCC C compiler, which is free and open source software.

3) Wireshark packet Tracer

Wireshark is a graphical user interface based FOSS that was developed for the analysis of network packets [18]. Besides being an open-source tool, Wireshark is a cross-platform software that runs on many platforms including Microsoft and GNU/Linux. Wireshark has the capability to understand the structure of different networking protocols, where it can analyze and display the fields of the protocols. Based on the above discussed Wireshark's capabilities, the tool has the potential for analyzing the received periodic Hello packets exchanged between nodes in a vehicular network. Authors in [19] used the Wireshark tool in their on-field experiments on vehicular obstructions in VANETs, in order to analyze the PDR and received signal strength indicator (RSSI).

B. Test Environment

1) Hardware:

The experiments have been performed using two sedan vehicles, each of which acts as a node in the vehicular network. Each vehicles is equipped with a laptop (as a data processing unit), a WiFi network adapter, an Omni directional antenna and a mouse GPS.

The laptops were running Ubuntu 14.04 operating system and were connected with the antenna to extend and improve the wireless range and performance. Since the antenna could not be directly connected to the laptop, a wireless USB network adapter was used as an interface between the two.

In order to find the position of the vehicles, the mouse GPS was used to read the latitude and longitude coordinates of the same.

2) Software programs:

An effective, flexible and reliable programming code plays a major role in yielding effective results during field experiments. The programming needed for performing experimental field tests on VANETs was accomplished by developing an efficient socket program. A socket is adopted to act as an Application Programming Interface (API) between two different and simultaneously running processes across an ad hoc network. In the experiments done, the two processes running include the process running at the sender end, and the other process running at the receiver end.

The type of socket used is UDP socket, which is connectionless datagram socket. Each of the packets, which is a “Hello” packet is broadcasted to every other node in the ad hoc network and receives packets from every node in the network. Since only two nodes have been used, the packets are uninterruptedly being broadcasted and received to and from one node to the other. In addition to the socket program written in C language, number of file handling

functions were also implemented to save the number of packets being sent and received at each node, in order to calculate the Packet Deliver Ratio of one node with respect to the other. Another important function deployed in the code was the calculation of the distance between the two nodes by using a mathematical haversine formula (1-3) based on the GPS coordinates of the receiver.

$$a = \sin^2(\Delta\varphi/2) + \cos\varphi 1 \cdot \cos\varphi 2 \cdot \sin^2(\Delta\lambda/2) \quad (1)$$

$$c = 2 \cdot \mathrm{atan2}(\sqrt{a}, \sqrt{(1-a)}) \quad (2)$$

$$d = R \cdot c \quad (3)$$

where φ is latitude, λ is longitude, R is earth's radius (mean radius = 6,371km).

With the reception of each Hello packet, the calculated distance between the two nodes is saved in a file along with the corresponding coordinates and MAC addresses of the sending node. This is further on used to test the accuracy of the distance measured by the GPS coordinates formula with respect to the actual distance, measured by a laser range finder (LRF). The difference in distance between the two discussed techniques is shown in section III.

The GPS receiver connected at each node receives information in the NMEA format, which is constantly saved into a file. Then the AWK scripting language is used to extract the location, and UTS time of the receiver node and the sender node.

The data being sent in each of the packets comprises of the location coordinates in latitude and longitude, the MAC address of the sender/receiver and the corresponding UTC time of the GPS clock, and the packet headers. The total size of each packet flowing through the network was 512 bytes. A total of 5000 packets were sent from both the nodes for each experiment, and the number of receiving packets is noted.

3) Field setup:

The nodes have been placed in a static (fixed) mobility mode scenario, each of them equipped with the devices mentioned in Table 1. The GPS was configured at each node to give the required NMEA format at a baud rate of 4800. For each of the experiments, all other parameters were kept fixed, and the distance between the two vehicles was being varied from the range of 10m to 450m, where distances were measured using aTruPulse 360BLRF. While measuring the distance, it was ensured that no obstacle, such as a vehicle or a building is present between the two nodes. Fig. 1 depicts the arrangement of the equipment listed in table 1. Each of the vehicles is installed with equipment in a similar fashion as shown in the figure.

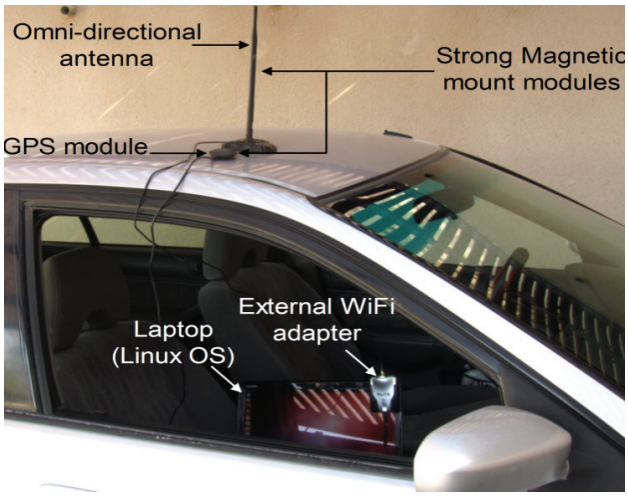


Fig. 1.Arrangement of equipment at each node (vehicle)

XLV. RESULTS

Two kinds of on-field experiments are performed in this work. The first experiment presents the accuracy of the measured inter-nodes distance,in a fixed nodes scenario, through the assistance of exchanged GPS coordinates. Conversely, the second experiment depicts the PDR fall as a function of distance from source.

In the first experiment, it was observed that the GPS-based measured inter-node distance did not produce a very significantly large error as compared to the LRF-based measured distance, as shown in Fig. 1.The experiments included inter-node distances varying from 10m to 450m.Fig. 1 presents both the magnitude of distance difference and the difference in percentage from the LRF-based measured distance.

The absolute positioning achieved by the GPS receivers can vary with time. This phenomenon is usually noticed due to a number of time varying effects. The effects includes the absence or availability of satellites and weather conditions controlling clarity of the sky, hence the line-of-sight (LOS) communication with the satellites. As a result of varying absolute positioning, the inter-node distance may vary for consecutively measured distances. However, as results depict in Fig. 1, the margin of error for all six experiemnts did not exceed more than 6%.

The second experiment shows the achieved PDR as a function of GPS-based distance. The distance is measured by the Haversine formula with assistance of the exchanged GPS coordinates. The coordinatesare recorded in each experiment with the considered varying distances. As shown in Fig. 2, the PDR followed a degrading trend as the distance increased. The observed phenomenon is as a result of increasing packet drops as distance increases from the broadcasting source.

The broadcasted messages have the probability of either being received or lost, which can be due to packets collisions, path loss and fading channels effects. The effects of packets collisions can be observed at any distance from the source. However, the effects of path loss and fading channel effects is more visible at larger distances from the source, as seen in Fig. 2. The PDR at a distance of 450m is approximately 28% in contrast to the 98% achieved at a distance of 10m. As observed in Fig. 2, a rapid fall in PDR is observed after distance of 250m from source indicating

that higher transmission power are required for achieving better PDR performance at distances over 250m.

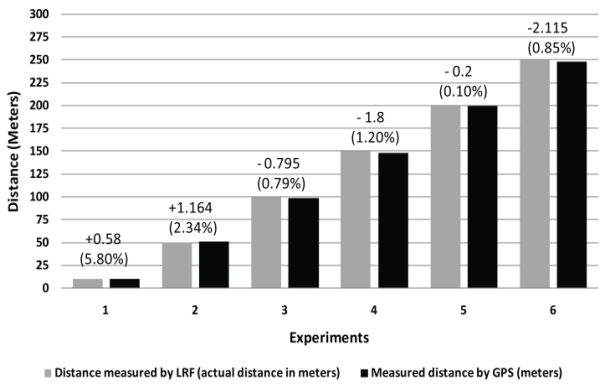


Fig. 2. Error margins in inter-node distance for fixed nodes in VANETs

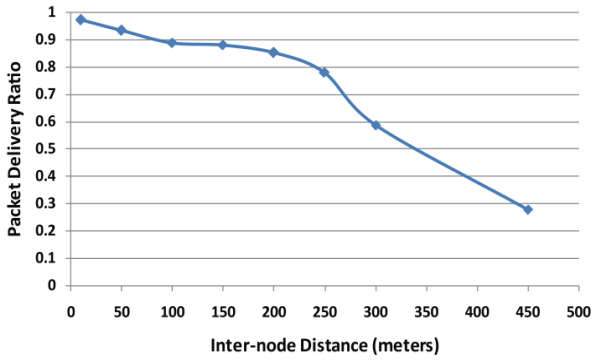


Fig. 3PDR as function of distance in VANETs on-road experiments

XLVI. Challenges & Future Work

Ashort coming of using FOSS tools was the GPS configuration with the Linux based system. Each time, the GPS dongle was mounted; it required a series of configuration steps to be followed before the GPS could receiver could start receiving signals from the GPS satellites and output an appropriate NMEA format output.

The future work would include calculating the average transmission delay of sending the packets over the network. And the node density will be increased to four nodes and further intensive tests will be performed in a dynamic mobility scenario with varying parameters including speed, packet size, and frequency of transmission.

XLVII. Conclusions

This paper has described the useof FOSS tools in implementing field tests for Vehicular Ad Hoc networks. These tools have proven to be a promising approach towards the implementation and provision of a considerably flexible and reliable method to validate and compare the results obtained through simulations of VANETs and deployment in actual road scenarios. Using FOSS tools in the VANET field experiments provides the user with flexibility to amend the software programs according to their requirements. Thus, the future of field tests in VANETs can be taken further ahead towards designing and testing efficient routing protocols and implementing several other road safety applications.

Acknowledgment

This paper is based upon research work supported by The Research Council (TRC) Oman, under Grant No: ORG/ICT/12/002, (www.trc.gov.om).

This research has used resources of the Free & Open Source Software (FOSS) Lab, under the Communication & Information Research Center (CIRC) at Sultan Qaboos University (SQU) - Oman, (www.squ.edu.om/circ/).

The authors would like to acknowledge the Information Technology Authority (ITA) - Oman, (www.ita.gov.om), for the support in establishing the FOSS lab at SQU.

References

[41] Jakubiak, Jakub, and YevgeniKoucheryavy. "State of the art and research challenges for VANETs." *Consumer Communications and Networking Conference, 2008*. CCNC 2008. 5th IEEE. IEEE, 2008.

[42] Toor, Yasser, Paul Muhlethaler, and AnisLaouiti. "Vehicle ad hoc networks: Applications and related technical issues." *Communications Surveys & Tutorials,* IEEE 10.3 (2008): 74-88.

[43] Qian, Yi, and Nader Moayeri. "Design of secure and application-oriented VANETs." *Vehicular Technology Conference*, 2008. VTC Spring 2008. IEEE. IEEE, 2008.

[44] Bansal, Meenakshi, Rachna Rajput, and Gaurav Gupta. "Mobile ad hoc networking (MANET): Routing protocol performance issues and evaluation considerations." *The Internet Society* (1999).

[45] Mihail L. Sichitiu et al. "Inter-Vehicle Communication systems: A Survey", *IEEE Communications surveys*, 2nd Quarter 2008, Volume 10, No.2. pp-88-105.

[46] Al-Sultan, S., Al-Doori, M. M., Al-Bayatti, A. H., &Zedan, H. (2014). A comprehensive survey on vehicular Ad Hoc network. *Journal of network and computer applications*, 37, 380-392.

[47] Fan Li and Yu Wang, "Routing in Vehicular Ad Hoc Networks: A Survey", *IEEE Vehicular Technology magazine*, JUNE 2007, University of North Carolina at Charlotte.

[48] Christian Wei, "V2X communication in Europe – From research projects towards standardization and field testing of vehicle communication technology", *Computer Networks*, Volume 55, Issue 14, 6 October 2011, Pages 3103–3119.

[49] Stubing, H. ;Bechler, M. ; Heussner, D. ; May, T., "simTD: a car-to-X system architecture for field operational tests [Topics in Automotive Networking]", *Communications Magazine, IEEE* (Volume:48 , Issue: 5), pp. 148 – 154, May 2010.

[50] Massimo Bertozzi, Alberto Broggi, Alessandra Fascioli, "Vision-based intelligent vehicles: State of the art and perspectives", *Robotics and Autonomous Systems* 32 (2000) 1–16.

[51] R. Stahlmann, A. Festag, A. Tomatis, I. Radusch, F. Fischer, "starting European field tests for car-2-x Communication: the drive c2x framework", *18th ITS World Congress and Exhibition 2011.*

[52] Buchenscheit, A., Schaub, F. ; Kargl, F. ; Weber, M., "A VANET-based emergency vehicle warning system", *Vehicular Networking Conference (VNC)*, 2009 IEEE, 28-30 Oct. 2009,pp. 1-8.

[53] Ubuntu Manpage: deb - Debian binary package format". Manpages.ubuntu.com. Retrieved 2013-07-05.

[54] Osama M. Hussain, HadjBourdoucen, and Mohamed Ould-Khaoua. "Impact of ns-2 as FOSS Simulation tool for research in vehicular ad-hoc networks." *Free & open source software conference (FOSSC-13).* Muscat, Oman. 2013.

[55] Love, R., Are, S. H. W., Linus, A. C., Kernels, L. V. C. U., & Begin, B. W. (2004). Linux Kernel Development Second Edition.

[56] Aho, Alfred V., Brian W. Kernighan, and Peter J. Weinberger. "Awk—a pattern scanning and processing language." Software: Practice and Experience 9.4 (1979): 267-279.

[57] http://www.csd.uoc.gr/~hy556/material/tutorials/cs556-3rd-tutorial.pdf

[58] G. Combs *et al*., "Wireshark-network protocol analyzer," [Online]. Available: http://www.wireshark.org

[59] Meireles, R.; Boban, M.; Steenkiste, P.; Tonguz, O.; Barros, J., "Experimental study on the impact of vehicular obstructions in VANETs," *Vehicular Networking Conference (VNC), 2010 IEEE* , vol., no., pp.338,345, 13-15 Dec. 2010.

A review of CRC mechanisms and FOSS algorithms

Mahdi Amiri-Kordestani[1], Afaq Ahmed[3] , HadjBourdoucen[2]

***Abstract*— The aim of this paper is to provide an overview for cyclic redundancy check method known as CRC mechanism applied to some popular FOSS applications. This is done through reviewing a number of notable papers published during the last fifty years. Major implementations and open source algorithms including bit-wise, byte-wise, lookup table, slicing-by-4 and 8 have been reviewed, studied and presented.**

Index Terms*— *CRC, Cyclic Redundancy Check, Fault tolerant, FOSS Algorithms

XLVIII. INTRODUCTION

Error checking codes are mechanisms to make sure data has been transmitted or stored correctly. In other words, they are used for investigating data integrity [14]. There are many different integrity mechanisms, from simple parity checks to advanced hash functions but Cyclic Redundancy Check (CRC) is a simple, fast, yet powerful common mechanism introduced by Peterson[11] in 1961 which is based on the remainder of a polynomial division in the finite field theory based on cyclic codes. Peterson introduced two Linear Feedback Shift Register circuit to implement CRC and his method is known as bit-wise CRC. Around 1983, byte-wise CRC algorithm was introduced by Perez[10] for faster calculation of CRC mechanism in computers by software. Latter, CRC Look-up tables were introduced by Ramabadran [12] and Sarwate[13] around 1988 which are still in use by embedded and general computing systems. An attempt towards Parallel CRC was started by Albertengo[2] in 1990 to produce a method to calculate CRC in parallel. Sometime later, Williams, Ross N. developed a generic and portable fast C algorithm on 1993 based on Look-up tables which is still in use at FreeBSD kernel operating system. After the nineties, there were many researches done on CRC but notable research by Castagnoli[4] and Koopman[7, 6] led the CRC to selecting better polynomials. Around 2010, Intel[5] and AMD introduced Carry-less Multiplication (CLMUL) which improved computation of CRC on generic computers. Nowadays CRC is used in many communication protocols like Ethernet, iSCSI, Bluetooth, ATM, mobile networks, synchronous data link control(SDLC), high level data link control (HDLC), IrDa, USB, MMC storage and etc

[1] Communication and Information Research Center (CIRC), Sultan Qaboos University, P.O. 17, Al-Khodh, Muscat-123, Sultanate of Oman (phone: 968-24142863; fax: 968-24144336; e-mail: amiri@squ.edu.om)

[2] Communication and Information Research Center (CIRC), Sultan Qaboos University, P.O. 17, Al-Khodh, Muscat-123, Sultanate of Oman (phone: 968-24143696; fax: 968-24144336; e-mail: hadj@squ.edu.om)

[3] Department of Electrical and Computer Engineering, College of Engineering, Sultan Qaboos University, P.O. 33, Al-Khodh, Muscat -123, Sultanate of Oman; (phone: 968-24141327; fax: 968-24413454; e-mail: afaq@squ.edu.om)

Terminology

The following abbreviations have been used in this paper:

- **G(x)**: Generator (used in CRC dividing) in form of a polynomial.
- **M(x)**: Message (a fixed size of data) in form of a polynomial.
- **R(x)**: CRC check bits in form of a polynomial.
- **T(x)**: CRC encoded message in form of a polynomial.
- **H(x)**: CRC encoded message with errors in form of a polynomial.
- **E(x)**: Error pattern in form of a polynomial.

XLIX. FIRST TITLE OF PARAGRAPH

A. Polynomial representation

The idea behind CRC comes from the well-known cyclic codes where the encoded data should be partitioned to fixed size blocks. For instance, consider each binary block as a message containing k bits of data. Hence, a block of 8 bits will have 8 bits of either 0 or 1. Each bit from the most important bit from left side (could be right) can be a coefficients for a term variable x^k where k is the position of the bit in the block starting from the first bit at the right in position 0 to the last bit at the left in position $k-1$. For instant, a block of 11110001 has 8 bits and represents an eight-term polynomial with coefficients of 1, 1, 1, 1, 0, 0, 0 and 1:

- High order to low order:
  $1x^7 + 1x^6 + 1x^5 + 1x^4 + 0x^3 + 0x^2 + 0x^1 + 1x^0$
- Low order to high order:
  $1x^0 + 1x^1 + 1x^2 + 1x^3 + 0x^4 + 0x^5 + 0x^6 + 1x^7$

The order in representing data as polynomials depends on the hardware, the way data is going to be transmitted, retrieved or stored and is described by term Endianness. In Big-Endian the most important byte (normally the byte from left hand side) is stored at lowest address where in Little-Endian, the least important byte is stored at the lowest address. Bit-Endians or network-order describes the way the binary data is going to be sent, Ethernet and RS-232 send the low bit first (Little-Endian).[14][11].

B. Mathematics of CRC

To simplify the mathematics and hardware implementations, CRC is using modulo 2 rules of algebraic finite field theory which is also known as Galois field (GF) in all arithmetic operations. In GF(2) theory, there are only two elements: *0* and *1* and to simplify the operations, the carry of adding and subtracting will be omitted, so: $1x^n + 1x^n = 0x^n$, $1x^n - 1x^n = 0x^n$, $1x^n = -1x^n$. Both addition and subtracting operations are equal to logical exclusive-or operation (XOR) which can be

simply implemented in hardware by using combination of shift registers and XOR gates.

In order to encode data in CRC, both parties (for example the sender and receiver) should first agree on a fixed polynomial called the generator G(x). For example the generator of CRC-4-ITU is the polynomial of $G(x) = x^4 + x + 1$. This polynomial has a degree of 4; where in binary format it is 10011 and has 5 bit length. In CRC terminology, CRC − n referrers to a polynomial of degree n; for instance, CRC − 32 which is used in Ethernet has a polynomial with degree of 32 with a maximum length of 33 bit length for its generator and 32 bit length for appended CRC check bits.

To generate the CRC encoded message, the first step is to portion data to fixed sized blocks. The block size length can be from one byte to several gigabytes. Each block of data is called $M(x)$ and represents a polynomial with a degree of $k-1$, where k is the length of the block in bits. The rest is to multiply $M(x)$ by x^r, where r is the degree of polynomial $G(x)$ and then divide the result by the generator $G(x)$. The remainder of such division is called $R(x)$ and is the CRC check bits of $M(x)$ by $G(x)$. The final step is to add the remainder $R(x)$ to $x^rM(x)$.

The calculation can be summarized in the following steps:

1. Data will be portioned to blocks of k bit length, each block will generate a message of $M(x)$ which is a polynomial of degree $k - 1$.
2. Calculation of $x^rM(x) = G(x)Q(x) + R(x)$ will be done on each block. Where r is the degree of the generator $G(x)$, and $Q(x)$ is the quotient which will be omitted and $R(x)$ is the remainder.
3. Encoded message will then be $F(x) = x^rM(x) + R(x)$.

As an example, assume a data with length of 1024 bits. If we want to use CRC-4-ITU with the generator of $G(x) = x^4 + x + 1$ which has a degree $r = 4$, we can divide the data to blocks of 8 bits. Suppose the first block contains 11110001 and the polynomial is $M(x) = x^7 + x^6 + x^5 + x^4 + 1$. Then $x^rM(x)$ will become $x^{11} + x^{10} + x^9 + x^8 + x^4$ and the remainder of dividing $x^rM(x)$ by $G(x)$ in modulo 2 would be $R(x) = x^2 + 1$. Finally the encoded message will have the following form:

$T(x) = x^rM(x) + R(x) = x^{11} + x^{10} + x^9 + x^8 + x^4 + x^2 + 1$

Figure 1 CRC Encoded Message

Additional explanation on CRC mathematics can be found in [12], [14] and [11].

C. Common Types of Errors

The novel aspect of CRC resides in its ability to detect common transmission errors consisting erroneous data. Assume one has sent the encode message T(x) and received the message with errors, $H(x)$ where $T(x) <> H(x)$. We can write $H(x) = T(x) + E(x)$ where $E(x)$ would be the error pattern. If datahad only one erroneous bit $(E(x) = x^i)$, the error is named "Single Error". But, if there are two erroneous bits $(E(x) = x^i + x^j$ where $i > j)$, the error is named "Double Error". Similarly we can define "Triple Error" where there are tree erroneous bits ($E(x) = x^i + x^j + x^k$ where $i > j > k$) and so on.

A "burst error" is a series of continues erroneous bits which can be defined as:

$$E(x) = x^m \sum_{0}^{n-1} x^i = x^{m+n-1} + x^{m+n-2} + \ldots + x^m$$

Where n is the number of erroneous bits, and m is the place of burst error from the right hand side.

Another interesting topic in data integrity is Hamming Distance (HD). In coding theory,HD shows the minimum number of erroneous bits which cannot be detected by a mechanism. For example, HD of one means there can be an undetectable single error. In general, HD of k means, there can be k erroneous bits in the message which are not detectable by mechanism while all combinations of $k - 1$ and less erroneous bits are detectable.

D. CRC Data Integrity

To check the integrity of data, the receiver divides T(x) by G(x):

$$\frac{T(x)}{G(x)} = \frac{x^rM(x) + R(x)}{G(x)}$$

As $x^rM(x)$ have already defined previously as $M(x) = G(x)Q(x) + R(x)$, so we will have:

$$\frac{T(x)}{G(x)} = \frac{G(x)Q(x) + R(x) + R(x)}{G(x)} = Q(x)$$

It's obvious if the remainder is zero $(R(x) = 0)$, the receiver will not detect any error butby choosing a good generator G(x), CRC mechanism can detect a variety of errors including single errors, double errors and some burst errors as well. We have the following operation in case of errors:

$$\frac{H(x)}{G(x)} = \frac{T(x) + E(x)}{G(x)} = \frac{x^rm(x) + R(x) + E(x)}{G(x)} = Q(x) + \frac{E(x)}{G(x)}$$

The remainder of dividing $H(x)$ by $G(x)$ will become zero only if the remainder of dividing $E(x)$ by $G(x)$ is also zero. For all detectable errors, this remainder should not be zero; otherwise the CRC mechanism will not detect it. In other words, for all undetectable errors one should have:

$$E(x) = Q_e(x)G(x).$$

In a single error case, we have $E(x) = x^i$, so we may have $x^i = Q_e(x)G(x)$. If a polynomial $G(x)$ contains two terms like x + 1, the equality cannot be correct which show that all polynomials $G(x)$ with more than one term (like $x + 1$), can identify single errors.

For double error case, we have $E(x) = x^i + x^j$ where $i > j$, so we may have:

$$x^i + x^j = Q_e(x)G(x)$$
$$x^j(x^{i-j} + 1) = Q_e(x)G(x)$$

And for all burst errors case, wewe'll have:

$$E(x) = x^m \sum_{0}^{n-1} x^i = Q_e(x)G(x)$$

As described earlier, by choosing a good polynomial for CRC generator, it can detect all burst errors less than the degree of its generator polynomial and can detect all errors on the odd number bits of the message. Some examples and mathematical proof on how CRC can detect these errors has been firstly introduced in [11] and also explained in a number of papers, e.g. [12] and [14].

L. CRC Implementation

Traditional hardware implementation of CRC was based on a simple shift register circuits that handled the data bit by bit. By using computers instead of computing bit by bit, it's much faster to work with bytes, words or double words. A faster solution is to have or create a look-up table in advance and use that table instead of real computing.

Different standards on CRC vary on size of message, generator polynomial, reversing the message or polynomial bits, an initial value for remainder and also an additional mask for the final remainder. Some of the common CRC versions are CRC-8, CRC-12, CRC-16 and CRC-32. Not all standards or telecommunication systems are using the same generator polynomials, for example CRC-8 has all the following popular polynomials:

- CRC-8 ($x^8 + x^7 + x^6 + x^4 + x^2 + 1$)
- CRC-8-ATM ($x^8 + x^2 + x + 1$)
- CRC-8-CCITT ($x^8 + x^7 + x^3 + x^2 + 1$)
- CRC-8-Dallas/Maxim ($x^8 + x^5 + x^4 + 1$)
- CRC-8-SAE J1850 ($x^8 + x^4 + x^3 + x^2 + 1$)

Probably the most famous CRC is CRC-32 which is used in the Ethernet. Researchers have shown that there are many alternate generator polynomials which could be used in CRC-32 method to improve the error detecting efficiency [7][1].

CRC-32C uses a better polynomial and was proposed to be used in iSCSI in RFC3385 around 2002. Nowadays, both Intel and AMD support SSE4.2 instructions in their modern chips and do the most of CRC-32C mechanism directly in hardware level with just a few lines of assembly code[5].

A. 3.1 Bit-wise Algorithm

CRC bit-wise (CRCB) algorithm is the simplest way to compute CRC and is based on a simple shift register circuit logic. For example, the following is a hardware implementation of polynomial $G(x) = x^4 + x + 1$ with LFSR (Linear Feedback Shift Register) including D-flip-flops and XOR gates. See Figure 2.

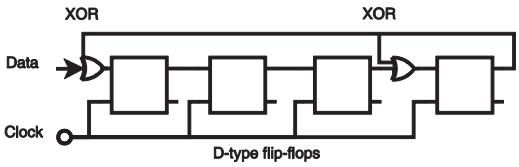


Figure 2: CRC-4-ITU with LFSR

The flowchart of CRCB is also very simple to understand and implement. Based on CRC's polynomial division, XOR operation will be executed between the message and generator starting from the left hand message bit (MSb) to the right hand bit (LSb) only if MSb is not zero (message is divisible). To achieve this, a simple loop will check the MSb first and if it is not zero, it will execute the XOR operation, then the message will be shifted left once, and if message is not finished yet, the loop repeats. See Figure 3.

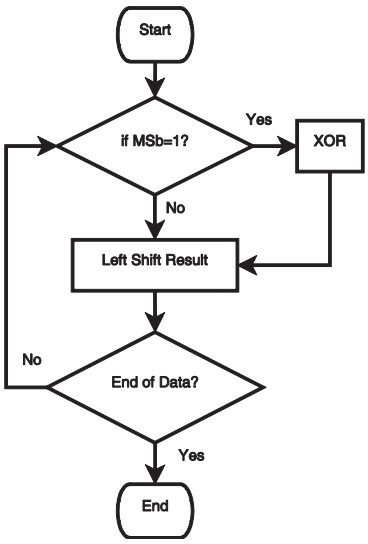


Figure 3: CRC Flowchart

CRCB can be easily implemented in C and other programming languages. The Algorithm 1will encode a single character with polynomial $G(x) = x^4 + x + 1$. The algorithm in using "unsigned char" for storing the polynomial value and can handle generator polynomials with maximum degree of 7. The code in Algorithm 1 is based on the code provided at [3] and is public domain.

```
#define POLYNOMIAL 0x98 /*10011000*/
unsigned char
crc(unsigned char const message)
 {
unsigned char remainder,bit;
remainder = message;
for (bit = 8; bit > 0; --bit)
  {
if (remainder & 0x80)
  remainder ^= POLYNOMIAL;
remainder = (remainder << 1);
  }
return (remainder >> 4);
}
```

Algorithm 1: CRC C source code

B. Lookup Table Algorithm

To archive faster computation of CRC mechanism, we can create a lookup table for all possible values of a message in advance and use the lookup table instead of recalculating CRC every time. For a message *M(x)* containing *k* bits of data, we will need a table with 2^k rows to store all possible

values. The Table 1 demonstrates the CRC check value of polynomial $G(x) = x^4 + x + 1$ for a single byte message.

Value	CRC Check
0	0
1	3
2	6
3	5
4	12
5	10
...	...
255	4

Table 1: CRC Lookup check bits (decimal)

We can easily create and manipulate the table by using a single array of size 2^k. The code in Algorithm 2 will create the table for all possible single byte messages (k = 8) by calling the function defined in Algorithm 1.

```
unsigned char crcTable[256];
voidcrcInit()
 {
for (inti=0;i<256;i++)
  crcTable[i]=crc(i);
};
```

Algorithm 2: CRC Lookup Table

By using array "crcTable" as a lookup table, to get the CRC remainder of M(x), we can easily read the value crcTable[M(x)]. More explanation and better algorithm can be found on [13] which is very popular in general and embedded systems.

C. More implementations

Although the Algorithm shown in 1 works fine on a single byte message, but one cannot load a very long message directly to registers in order to calculate the XOR value & we need to slice the data to smaller parts. Assume $M_L(x)$ is a long message which we can write it as $M_L(x) = M_1(x):M_2(x)$. Here the ":" operation means we have sliced $M_L(x)$ is two parts $M_1(x)$ and $M_2(x)$. We can say:

$$[M_1:M_2] \bmod G = [x^k M_1(x) + M_2(x)] \bmod G = [x^k(Q_1(x)G(x) + R_1(x)) + M_2(x)] \bmod G = [x^k Q_1(x)G(x)] \bmod G + [x^k R_1(x) + M_2(x)] \bmod G = [R_1:M_2] \bmod G$$

Based on the theorem above, one can divide long data to smaller pieces of data and calculate the CRC of each part individually. This is the main idea behind faster implementations of CRC and parallel CRC approaches. Two modern implementations are the slicing-by-4 and slicing-by-8 algorithms from [9] which have a significant speed improvements compared to previous works. Figure 3 shows the original proposed algorithm in [9].

```
crc = INIT_VALUE;
while(p_buf<p_end ) {
crc ^= *(uint32_t *)p_buf;
 term1 = table_56[crc& 0x000000FF] ^
   table_48[(crc>> 8) & 0x000000FF];
 term2 = crc>> 16;
crc = term1 ^ table_40[term2 &
   0x000000FF] ^ table_32[(term2 >> 8)
& 0x000000FF];
p_buf += 4;
}
returncrc^FINAL_VALUE;
```

Algorithm 3: Slicing by 4 from [9]

D. CRC in operating systems

1) Linux

The Linux operating system is the most popular open source operating system for all embedded, mobile, server and desktop usages. At the Linux kernel library directory, there are many implementations for CRC including CRC-7, CRC-8, CRC16, CRC-CCITT, CRC-ITU-T and during compile time, the main algorithm of CRC-32 can be chosen to one of the following algorithms:

1. Slice-By-8 (the default) [9]
2. Slice-By-4 [9]
3. Sarwate's Algorithm [13](one byte at a time)
4. Classic one Algorithm (one bit at the time)

Algorithm 4 is located in "/lib/xz/xz_crc32.c" and is used in XZ decompression.

```
XZ_EXTERN uint32_t
 xz_crc32(const uint8_t *buf,
size_t size, uint32_t crc)
{
crc = ~crc;
while (size != 0) {
crc = xz_crc32_table[*buf++ ^
    (crc& 0xFF)] ^ (crc>> 8);
 --size;
 }
return ~crc;
}
```

Algorithm 4: Part of Linux Kernel (CRC-32)

2) FreeBSD

The FreeBSD operating system is one of the oldest operating systems with a friendly license for commercial use. Algorithm 5 is the CRC-32 function within the kernel source code (sys/libkern/crc32.c). You can also find implementations for CRC-32C and slicing-by-8 within the same file. This algorithm is using "crc32_tab" as a predefined look-up table for CRC computations.

```
uint32_t
crc32(const void *buf, size_t size)
{
const uint8_t *p = buf;
uint32_tcrc;
crc = ~0U;
while (size--)
crc = crc32_tab[(crc ^ *p++) &
```



```
        0xFF] ^ (crc>> 8);
returncrc ^ ~0U;
}
```

Algorithm 5: Part of FreeBSD Kernel (CRC-32)

LI. CONCLUSION

The aim of this paper was to provide a brief review for cyclic redundancy check method known as CRC mechanism applied to some popular FOSS applications. Based on the works of [1, 7, 6, 9] there are still many investigation research areas for improving these mechanisms. The main efforts are directed to improve the characteristics of CRC by finding better implementations and more efficient polynomial generators to improve CRC efficiency especially for large amounts of data transfers in FOSS and other applications.

ACKNOWLEDGMENTS

This research was carried out at the Free & Open Source Software (FOSS) Lab, under the Communication & Information Research Center (CIRC) at Sultan Qaboos University (SQU) - Oman, (www.squ.edu.om/circ/). The researchers would like to acknowledge the support from Information Technology Authority, ITA, for providing the needed resources.

REFERENCES

[1] A Ahmad, L Hayat, "Selection of Polynomials for Cyclic Redundancy Check for the Use of High Speed Embedded: An Algorithmic Procedure", W. Trans. on Comp., vol. 10, no. 1, pp. 16—20, 2011. URL http://dl.acm.org/citation.cfm?id=2001170.2001173.

[2] Guido Albertengo, Riccardo Sisto, "Parallel CRC Generation", IEEE Micro, vol. 10, no. 5, pp. 63—71, 1990.

[3] Michael Barr, "Slow and steady never lost the race", Embedded Systems Programming, vol. , no. , pp. 37—46, 2000.URL http://www.barrgroup.com/Embedded-Systems/How-To/CRC-Calculation-C-Code.

[4] G Castagnoli, S Brauer, M Herrmann, "Optimization of cyclic redundancy-check codes with 24 and 32 parity bits", Communications, IEEE Transactions on, vol. 41, no. 6, pp. 883—892, 1993.

[5] Vinodh Gopal, ErdincOzturk, James Guilford, WajdiFeghali, "Choosing a CRC polynomial and associated method for Fast CRC Computation on Intel Processors". 2012. URL http://www.intel.com/content/dam/www/public/us/en/documents/white-papers/fast-crc-computation-paper.pdf.

[6] P Koopman, T Chakravarty, "Cyclic redundancy code (CRC) polynomial selection for embedded networks". 2004.

[7] P Koopman, "32-bit cyclic redundancy codes for Internet applications". 2002.

[8] M E Kounavis, F L Berry, "A systematic approach to building high performance software-based CRC generators". 2005.

[9] Michael E Kounavis, Frank L Berry, "A Systematic Approach to Building High Performance Software-Based CRC Generators", 2013 IEEE Symposium on Computers and Communications (ISCC), vol. 0, no. , pp. 855—862, 2005.

[10] Aram Perez, "Byte-Wise CRC Calculations", IEEE Micro, vol. 3, no. 3, pp. 40—50, 1983.

[11] W W Peterson, D T Brown, "Cyclic Codes for Error Detection", Proceedings of the IRE, vol. 49, no. 1, pp. 228—235, 1961.

[12] T V Ramabadran, S SGaitonde, "A tutorial on CRC computations", Micro, IEEE, vol. 8, no. 4, pp. 62—75, 1988.

[13] D V Sarwate, "Computation of Cyclic Redundancy Checks via Table Look-up", Commun. ACM, vol. 31, no. 8, pp. 1008—1013, 1988.

[14] Andrew S. Tanenbaum, David J. Wetherall, Computer Networks. Prentice Hall, 2010.

SOME COMMON CRC POLYNOMIALS

The following list includes some of the common CRC polynomials and what they are known for. [34]

- CRC-1: The simplest form of CRC for parity checking. Polynomial is:$x + 1$
- CRC-4-ITU Used in ITU G.704, the polynomial is: $x^4 + x + 1$
- CRC-5-ITU Used in ITU G.704, the polynomial is: $x^5 + x^4 + x^2 + 1$
- CRC-5-USB Used in USB, the polynomial is: $x^5 + x^2 + 1$
- CRC-6-ITU Used in ITU G.704, the polynomial is: $x^6 + x + 1$
- CRC-7 Used in Telecom systems and Multimedia Card (MMC). Polynomial is:$x^7 + x^3 + 1$
- CRC-8 General, Polynomial is: $x^8 + x^7 + x^6 + x^4 + x^2 + 1$
- CRC-8-ATM Used in ATM HEC, Polynomial is: $x^8 + x^2 + x + 1$
- CRC-8-CCITT Used in 1-Wire bus, Polynomial is: $x^8 + x^7 + x^3 + x^2 + 1$
- CRC-8-Maxim Used in 1-Wire bus, Polynomial is: $x^8 + x^5 + x^4 + 1$
- CRC-8-SAE Used in SAE J1850, Polynomial is: $x^8 + x^4 + x^3 + x^2 + 1$
- CRC-10 General, Polynomial is: $x^{10} + x^9 + x^5 + x^4 + x + 1$
- CRC-12 Used in Telecom systems, Polynomial is: $x^{12} + x^{11} + x^3 + x^2 + x + 1$
- CRC-15-CAN Used in CAN, Polynomial is: $x^{15} + x^{14} + x^{10} + x^8 + x^7 + x^4 + x^3 + 1$
- CRC-16 Used in USB. Polynomial is: $x^{16} + x^{15} + x^2 + 1$
- CRC-16-CCITT Also known as CRC-CCITT and CRC-IBM, used in HDLC, SDLC, XMODEM, X.25, V.41, Bluetooth, PPP, IrDA, Polynomial is: $x^{16} + x^{12} + x^5 + 1$
- CRC-24-Radix64 General, Polynomial is: $x^{24} + x^{23} + x^{18} + x^{17} + x^{14} + x^{11} + x^{10} + x^7 + x^6 + x^5 + x^4 + x^3 + x + 1$
- CRC-32 Also known as CRC-IEEE, used in Ethernet, AAL5 (ATM Adaptation Layer 5) , FDDI (Fiber Distributed Data Interface) . Polynomial is: $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$
- CRC-32CKnow as CRC-32C or Castagnoli, used in iSCSI.Polynomial is: $x^{32} + x^{28} + x^{27} + x^{26} + x^{25} + x^{23} + x^{22} + x^{20} + x^{19} + x^{18} + x^{14} + x^{13} + x^{11} + x^{10} + x^9 + x^8 + x^6 + 1$
- CRC-32KKnow as CRC-32K or Koopman, Polynomial is: $x^{32} + x^{30} + x^{29} + x^{28} + x^{26} + x^{20} + x^{19} + x^{17} + x^{16} + x^{15} + x^{11} + x^{10} + x^7 + x^6 + x^4 + x^2 + x + 1$
- CRC-64-ISO Defined in ISO 3309, Polynomial is: $x^{64} + x^4 + x^3 + x + 1$

About a FOSS for the Teaching of FORTRAN Programming Language Course

Afaq Ahmad[1], Tariq Jamil[1], Saad Ahmad[2] and Amer Arshad Abdulghani[1]

***Abstract*— The FORTRAN programming language has its revival phase. Majority of the science colleges of the world universities are teaching FORTRAN programming as a primary language. At Sultan Qaboos University, FORTRAN programming language is taught using the Free Open Source Software (GFORTRAN compiler with Geany editor) since 2009. We share our knowledge about this Free Open Source Software for the benefits of scientists and engineers.**

***Index Terms*— FORTRAN, Geany, GFORTRAN, High Level Language, Scientific Language, Support file-types, Plugins.**

LII. INTRODUCTION

FREE Open Source Software (FOSS) drive a free software is program that provides users a certain levels of essential freedoms. Those freedoms are of kinds as:

- The freedom to run the program,
- The freedom to study how the program works and
- The freedom to redistribute copies of the programs as it is or in the modified forms.

Typical free software licenses such as the GPL, LGPL, Mozilla, Apache, MIT-style and BSD-style licenses grant these above described freedoms [1].

If a program fails to serve with these freedoms is termed as non-free program. Non-free program approach is unethical because it makes such program an instrument of unjust power. The issue of Free Software is not just a technical issue; it is an ethical, social, and political issue. It is related to an indirect question of the human rights that the users of software ought to have [1]. It is also to be reminded wit ourselves that freedom and cooperation are essential values of Free Software [1]. The freedom provided by Free Open Software Source plays a fundamental role in education in the forms of enhancement of learning processes and in obtaining fruitful research results. Hence to disseminate human knowledge and to prepare students to be good members of their community we should use and teach Free Software in educational institutions of all levels. The Free Software is the only software that allows them to accomplish their essential missions of achieving goals of education. The source code and the methods of Free Software are part of human knowledge. On the contrary, proprietary software is secret, restricted knowledge, which is the opposite of the mission of educational institutions. Free Software supports education, proprietary software forbids education [1] – [2].

To this direction we moved towards FOSS and developed many laboratory experiments [3] – [22] along with teaching of FORTRAN language programming.

LIII. FORTRAN LANGUAGE HISTORICAL VIEW

FORTRAN (Formula Translation) was originally developed in the 1950s, is one of the most popular languages in the field of numerical and scientific computation, and especially in High Performance Computing (HPC). Among the earliest High Level Languages (HLL) FORTRAN played a key role in computation, and is still in widespread use today. FORTRAN 66 released in 1966 was the first of all programming language standards. Hence, after the release of the first standard the following rapid developments came in existence.

- Second standard was released in 1978 as FORTRAN 77 and
- Third standard was released in 1991 as FORTRAN 90, by adding new, modern features such as structured constructs, array syntax and ADT.

FORTRAN 90 was extended and revised in 1997 and commercialized as FORTRAN 95. The release version of FORTRAN 95 was further, extended with published report in 2004, and marked as FORTRAN 2003, existed as 4th standard of FORTRAN language. Major revision, incorporates TRs, adds many new features like Object Oriented (OO) programming. The fifth standard of FORTRAN was released in 2010 and available as FORTRAN 2008. For more information about several versions of the FORTRAN language, interested readers can see reference [1], [23] – [25].

Although many versions of FORTRAN but FORTRAN 90 was a significant revision to the original standard. Newer versions do exist (FORTRAN 95/2003/2008), but they generally build upon the FORTRAN 90 core.

LIV. REVIVAL OF FORTRAN LANGUAGE

Considered the Latin of the computer programming world, FORTRAN, a workhorse of the last century, is today considered to be a forgotten language which is almost dead. An attempt made in 2003 to revive this workhorse by approving FORTRAN 2003 standard which incorporates features of object-oriented programming, IEEE 754 standard for floating point arithmetic, and inter-operation with the C

[1]Department of Electrical and Computer Engineering, College of Engineering, Sultan Qaboos University, P.O. 33, Al-Khodh, Muscat -123, Sultanate of Oman; (phone: 968-24141327; fax: 968-24413454; e-mail: afaq@squ.edu.om)

[2]Department of Electrical and Computer Engineering, Cockrell School of Engineering, The University of Texas at Austin, Austin, TX 78712, United States; (e-mail: ahmad8@utexas.edu)

programming language, has given new life to this beast. It is being used today as a tool of choice for carrying out very large numerical computations, such as simulating atomic bomb explosions, weather prediction, flight simulations, DNA proteins simulations, and electron-orbiting simulations. Programmers call FORTRAN a horrific language but, in the face of no other language having optimized compilers to suit computational tasks, the availability of vast sub-routines library of this language containing optimized code makes it attractive to the scientists [26], [27]. A table detailing the features of available FORTRAN compilers can be found at [28]. A more modern version of FORTRAN, approved in 2008, includes features for parallel programming and new capabilities are being added constantly to keep it up-to-date with the features found in other popular high-level programming languages and as demanded by the programmers worldwide [29].

Despite the fact that it is journey of over 63 years and serving professionally over more than 40 years, FORTRAN is still taught at many Universities to students of Physics, Chemistry, Engineering and more as their first ever formal introduction to programming in spite of existence of much better alternatives, such as C++, Java, Python that would serve a physical science undergraduate much better. Yes, it is true that at a stage when FORTRAN 77 was in use and the C++ programming has evolved, naturally, there was no comparison of C++ for its excellence with respect to FORTRAN 77. The initial research projects which were started during FORTRAN 77 are still continuing with upgraded FORTRAN 90/95/2003 and 2008. Some of such global projects are in computational physics, earth sciences, and agriculture for clients which include Departments of Energy and Departments of Defense laboratories as well as commercial clients as diverse as hazardous waste companies and grain companies. Programs for numerical methods and codes based on them for modeling contaminant flow in porous media, modeling the structural mechanics of resonant sonic drilling rigs, modeling bi-static ground penetrator radar propagation in partially saturated soils, modeling solid dynamics at high strain rates, modeling the phonological development of corn and soy beans, modeling solar insolation in the photo synthetically active spectrum based on first principle atmospheric physics, and developed neural networks for the detection of dust clouds from satellite imagery [30].

FORTRAN has strict aliasing semantics compared to C++ and has been aggressively tuned for numerical performance for decades. Algorithms that uses the CPU to work with arrays of data often have the potential to benefit from a FORTRAN implementation [30].

LV. GFORTRAN COMPILER AND GEANY EDITOR

The FORTRAN compiler (gFortran) has gained support for the IEEE intrinsic modules specified by the FORTRAN 2003 and FORTRAN 2008 standards. The code was contributed by François-Xavier Coudert of CNRS [30].

Amongst many available Fortran 90 compilers, perhaps the most common is gFortran (part of the GNU Compiler Collection). It is Free and Open Source Software. Using Geany, an Integrated Development Environment (IDE) for writing and compiling codes is a good combination with gFortran and provides many nice features.

The gFortran installer can be downloaded from the GFortran page [ ]. Geany, which is a text editor using the GTK2 toolkit with basic features of an integrated development environment can be downloaded from the Geany website [30].

The main programming languages that are included with the Raspberry Pi are Scratch and Python. Those who want to learn how to use other programming languages can do so by using a code editor. A useful code editor to have is Geany. Once the Geany program has installed it will appear in the menu options under programming. To look at the programming that can be accomplished on the Geany Text Editor launch the code editor from the menu. Next, go to the Document menu. The drop down menu has option for set file type. Click on this option on the menu you will see four active options.

- Programming Languages - this includes assembler, Pascal, Fortran, Java, C, C++
- Scripting Languages - Python, JavaScript, PHP, Perl, Ruby being the most commonly used in "real" programming
- Markup Languages - HTML, XML as examples
- Miscellaneous - SQL file types, YAML are available among others

The list above is an indication of the languages that Geany is able to help. To program in any of the languages involves typing the program in the text-editing window. By saving the program using the appropriate file extension such as in the case of Python .py, Geany will run the program as a python program.

Geany will not teach you programming but will allow you to experiment with Python and other languages all in the same basic text-editing environment. As indicated above you can also use Geany to program in Python. The IDE (Integrated Development Environment), which is what Geany and IDLE (on your Raspberry Pi) are, is a step up from using the LX Terminal and Leaf pad. Geany is more sophisticated than IDLE in what can be done in the same code editor.

A good example of how to use the Geany code editor for Python can be found in [30]. The book describes about using the Geany code editor and tries to impart some basic programming skills as well.

Geany is useful to have on the Raspberry Pi for the reason that it opens up some of the more common languages used in producing commercial software. The cross-platform nature of Geany allows you to use the Raspberry Pi as a development tool for say Windows 8 apps that combine HTML, CSS and JavaScript. Google is Python Powered. The Google App Engine for building courses Course builder is a good example of Python in action. Quite along blog post today about the merits of Geany. The Raspberry Pi has limits only determined by your imagination (ok, a little bit to with the hardware). If your program runs well on a Raspberry Pi it is bound to run well when you go cross platform.

Geany is currently developed by the following people: Colomban Wendling, Nick Treleaven, Matthew Brush, Enrico Tröger, Frank Lanitz and many other contributors, translators and patch writers. The regular contributors are Lex Trotman and Eugene Arshinov [30].

Geany is a small and lightweight Integrated Development Environment. It was developed to provide a small and fast IDE, which has only a few dependencies from other

packages. Another goal was to be as independent as possible from a special Desktop Environment like KDE or GNOME - Geany only requires the GTK2 runtime libraries. Some basic features of Geany are as below [30] – [32]:

- Syntax highlighting
- Code folding
- Symbol name auto-completion
- Construct completion/snippets
- Auto-closing of XML and HTML tags
- Call tips
- Symbol lists
- Code navigation
- Build system to compile and execute your code
- Simple project management
- Plugin interface

Geany supports file-types various kinds of programming languages. Table 1 provides the list of such programming languages.

Table 1: Geany support file-types

Abaqus	F77	Pascal
Abc	Ferite	Perl
ActionScript	Forth	PHP
Ada	Fortran	Po
Asciidoc	FreeBasic	PowerShell
ASM	Genie	Python
Batch	GLSL	R
C	Go	reStructuredText
C#	Graphviz	Ruby
C++	Haskell	Rust
CAML	Haxe	Scala
Clojure	HTML	Sh
CMake	Java	SQL
COBOL	Javascrip	Tcl
Conf	LaTeX	Txt2tags
CSS	Lisp	Vala
CUDA	Lua	Verilog
Cython	Make	VHDL
D	Markdown	XML
Diff	Matlab	YAML
Docbook	NSIS	
Erlang	Objective-C	

As for as plugins are concerned, the Geany provides followings:

- Classbuilder: Creates source files for new class types
- Export: Exports the current file into different formats
- Filebrowser: Adds a file browser tab to the sidebar
- HTML Characters: Inserts HTML character entities like '&'
- Save Actions: Provides different actions related to saving files (autosave, instantsave, backupcopy)
- Split Window: Splits the editor view into two windows

In general, it is said that Geany should run on every platform, which is supported, by the GTK libraries. Geany comfortably runs in environments of various operating systems such as:

- Linux,
- FreeBSD,
- NetBSD,
- OpenBSD,
- MacOS X,
- Solaris Express and
- Windows

Here, it is to be noted that only the Windows port of Geany is missing some features such as AIX v5.3.

The code is licensed under the terms of the GNU General Public License with following available versions.

- Ubuntu 13.10
- Ubuntu 13.04
- Ubuntu 12.10
- Ubuntu 12.04
- Ubuntu 11.10
- Ubuntu 11.04
- Ubuntu 10.04

LVI. CONCLUSION

FORTRAN 77 was a terribly outdated language due to this reason C++ has attracted the attention it has in the scientific community. FORTRAN 90 has every feature in C that is important to scientific programming and most of the features of an object-oriented language. FORTRAN 90 lacks only inheritance and that was resolved by FORTRAN 2000.

FORTRAN 90 is designed to generate executable codes that are highly optimized and thus run extremely fast unlike C and C++. An example is pointers. Pointers are integral to C and C++ programming and because the compiler cannot determine whether a pointer is aliased, it is impossible for it to determine inter procedural dependencies. The result is a significant degradation in optimization and extremely slow execution speeds for most scientific codes with C and C++.

FORTRAN 90 pointers are designed to give the functionality of pointers, but with restrictions that eliminate issues such as aliasing. From a programming perspective however, an even more important point is that FORTRAN 90 has more natural ways of expressing the functionality that C and C++ require pointers to express. Because of this, FORTRAN 90 90 is a more natural language to program in and the time required for debugging codes is a fraction of that required by C and C++ (C++ is much worse than C).

FORTRAN 90 has another major advantage over C or C++ toward the programming parallel computers, which is in terms of linear memory model, only FORTRAN 90, has addressed this problem and providing standardized language support for parallelism. This support includes array syntax and many intrinsic functions for doing array operations varying from reduction operations such as array sums to matrix operations. With the use of FORTRAN 90 operator overloading and polymorphism, one can significantly extend the number of operations that avoid any reliance on the linear memory model.

We authors think that it is reasonable that students must learn C++ before they graduate, though even more important is that they learn how to program MATLAB. However, the issue is what freshmen should learn as their first language and for that, we would like to recommend FORTRAN 90 hands down. This is because it is a better language for scientific programming and is both easier to learn and use than the alternatives.

ACKNOWLEDGMENT

The authors are thankful to Sultan Qaboos University, Sultanate of Oman for providing research facilities, technical supports and research environment. The authors would like to express their great appreciations and gratitude for this to the university.

REFERENCES

[1] http://www.gnu.org/education/education.html
[2] Cambridge online learning community cambolc.co.uk
[3] Ahmad, A., Rizvi, M. A. K., Al-Lawati, A., Al-Abri, D. and Awadalla, M., "Design of a Probabilistic Based Software Tool for evaluating Controllability, Observability and Testability Models of Digital Systems", Indian Journal of Science and Technology (IJST), vol. 7, no. 10, pp. 1525- 1537, 2014.
[4] Samir A. Al-Busaidi, "Complementing Digital Logic Design with Logisim", TJER Vol. 11, No. 1, pp. 69-76, 2014.
[5] Ahmad, A., Ahmad, S., Al-Habsi, A. and Al-Hinai, M., (2014), "Understanding Universal Product Code – A Study and Simulation Experiments", Indian Journal of Industrial and Applied Mathematics (Taylor & Francis) (IJIAM), vol. 5, no. 1, pp. 25 – 34, 2014.
[6] Samir A. Al-Busaidi, "Complementing Digital Logic Design with Logisim", Free and Open Source Software Conference" (FOSSC-13), held at Muscat, Oman, February 18-19, 2013, pp. 4-9.
[7] Ahmad, A., Al-Abri, D. and Bait-Shiginah, F., (2013) "Software Testing Education – A Required Knowledge for Software Systems' Developers and Managers" Free and Open Source Software Conference" (FOSSC-13), held at Muscat, Oman, February 18-19, 2013, pp. 29-33.
[8] Ahmad, A. and Ruelens, D. and S. Ahmad, (2013), "Development of verification tool for minimal Boolean equation", IEEE Technology and Engineering Education (ITEE), vol. 8, no. 4, pp. 1-6, 2013.
[9] Afaq Ahmad, Sayyid Samir Al-Busaidi, Mufeed Juma Al-Musharafi, (2013), "On Properties of PN Sequences generated by LFSR – a Generalized Study and Simulation Modeling", Indian Journal of Science and Technology (IJST), vol. 6, no. 10, pp. 5351-5358, 2013.
[10] Ahmad, A. and Ruelens, D., (2013),"Development of digital logic design teaching tool using MATLAB & SIMULINK", IEEE Technology and Engineering Education (ITEE), vol. 8, no. 1, pp. 7-11, 2013.
[11] Ahmad, A., Al-Abri, D. and Al-Busaidi. S. S., (2012), "Adding Pseudo-Random Test Sequence Generator in the Test Simulator for DFT Approach, Computer Technology and Applications", vol. 3, no. 7, pp. 463-470, 2012.
[12] Ahmad, A. and Al-Abri, D., (2012), "Design of a Pseudo-Random Binary Code Generator via a Developed Simulation Model", International Journal on Information Technology (ACEEE - Journal), vol. 2, no. 1(March Issue), pp. 33-36, 2012.
[13] Ahmad, A., and Bait-Shiginah, F., (2012) "A Nonconventional Approach to Generating Efficient Binary Gray Code Sequences", IEEE Potentials, vol. 31, no. 3, pp. 16-19, 2012.
[14] Ahmad, A., Dawood Al-Abri, (2010), "Design of an Optimal Test Simulator for Built-In Self-Test Environment", Journal of Engineering Research, vol. 7, no. 2, pp. 69 – 79, 2010.
[15] Ahmad, A., Al-Mashari, A., Al-Lawati, A., (2010), "Development of a Computer Based Syndrome Diagnostic Testing Tool to Help in Children's Learning Process", International Journal of Education and Development using ICT (IJEDICT), vol. 6, no.1, pp. 76 - 87, 2010.
[16] Ahmad, A., and Mohammed M. Bait Suwailam, (2009), "A Less Complex Algorithmic Procedure for Computing Gray Codes", The Journal of Engineering Research, vol. 6, no. 2, , pp. 12 -19,2009.
[17] Ahmad, A., (2007), "Another Perspective in Generation and using of Gray Code-words", Journal of Electrical Engineering, IEEE Malaysia, (ELEKTRIKA), vol. 9, no. 2, 2007, pp. 49 – 55.
[18] Ali Al-Lawati and Ahmad, A., (2004), "Realization of a simplified controllability computation procedure – A MATLAB-SIMULINK based tool", Sultan Qaboos University Journal for Scientific Research - Science and Technology, Oman, vol. 8, pp. 131 – 143, 2004.
[19] Ahmad A., Al-Lawati, A. M. J. and Ahmed M. Al-Naamany, (2004), "Identification of test point insertion location via comprehensive knowledge of digital system's nodal controllability through a simulated tool", Asian Journal of Information Technology (AJIT), vol. 3, no. 3, pp. 142 – 147, 2004.
[20] Ahmad A., (2003), - An Invited Paper - "Secrets of error detection scheme of ISBN system," Caledonian Tech-News – A Journal: Caledonian Journal of Engineering (CJE), vol. 1, no. 1, pp. 10 – 14, 2003.
[21] Ahmad A., Al-Musharafi, M. J., and Al-Busaidi, S., (2002), "Study and implementation of properties of m-sequences in MATLAB-SIMULINK – A pass / fail test tool for designs of random generators", S.Q.U. Journal of Scientific Research – Science and Technology, vol. 7, part 1, pp. 147 – 156, 2002.
[22] Ahmad A. and Elabdalla A. M., (1997), "An efficient method to determine linear feedback connections in shift registers that generate maximal length pseudo-random up and down binary sequences", Computer & Electrical Engineering - An Int'l Journal (USA), vol. 23, no. 1, pp. 33-39, 1997.
[23] http://www.cmap.polytechnique.fr/~barbier/Guide_fortran/append-d.html
[24] Wikipedia Fortran article
[25] http://en.wikipedia.org/wiki/Fortran
[26] http://blog.profitbricks.com/top-integrated-developer-environments-ides/
[27] http://c2.com/cgi/wiki?DeadLanguageFortran
[28] http://stackoverflow.com/questions/146159/is-fortran-faster-than-c
[29] http://www.polyhedron.com/fortran-compiler-comparisons/compilerfeatures
[30] http://www.hpcwire.com/2011/09/20/why_fortran_still_matters/
[31] http://www.geany.org/
[32] http://www.geany.org/manual/reference/
[33] http://plugins.geany.org/

FOSSC 2015
OPPORTUNITIES & COMMUNITIES