The OpenOSS Project – an open source operations support lab for the telecommunications industry

Craig GALLEN
cg02r@ecs.soton.ac.uk
Jeff REEVE
jre@ecs.soton.ac.uk
School of Electronics and Computer Science, University of Southampton
Hampshire, England, SO171JD

ABSTRACT
This paper discusses the aims, objectives and early deliverables from the OpenOSS project which has been set up with the sponsorship of a number of Telecommunications Service Providers to investigate the potential value of open source as an enabler for rapid innovation within the telecommunications industry.

This paper also reports on the success of an initial proof of concept of OpenOSS which was publicly demonstrated to the telecommunications industry in Nice at the 2005 Telemanagement World conference. We consider the practical business and technical benefits and challenges of using open source development in this way and report on our plans to take the project to the next stage of development.

Keywords: Open Source, NGOSS, OSS/J, Action Research, OSS, Operations Support Systems, Systems Integration, Requirements Management.

1. INTRODUCTION
This paper discusses the aims, objectives and early deliverables from the OpenOSS project which has been set up with the sponsorship of a number of Telecommunications Service Providers to investigate the potential value of open source as an enabler for rapid innovation of Operational Support Systems within the telecommunications industry.

Because the open source research community may not be familiar with the telecommunications industry, we begin with a brief introduction to Operational Support Systems before considering the role of the Telemanagement Forum’s standardisation activities.

We then consider the state of adoption of Open Source in this market segment and introduce the OpenOSS project which is seeking to use Open Source development to facilitate research into next generation network management architectures.

We describe the work done for an initial proof of concept and finally discuss our proposed next steps.

2. TELECOMMUNICATIONS OSS AND NGOSS
In the Telecommunications industry, Operations Support Systems (OSS1) enable the business processes through which Service Providers (SP’s) create a business out of running their networks. As an example we know of a typical national service provider which has over 1400 such systems integrated into a complex architecture. The process of introducing a new network technology or services requires significant engineering lead time to make changes across these systems and represents a major cost and loss of agility for the company. This story could be repeated in every major operator in the industry.

The Telemanagement Forum (TMF) [1] is the leading telecommunications industry body developing common standards for Telecommunications OSS. They are presently finalising a set of standards called New Generation Operations Support Systems (NGOSS) for the management of next generation networks [2]. NGOSS has four key components

- SID: Shared information and Data model which describes the information shared by OSS systems used by a service provider.
- eTOM: extended Telecoms Operations Map which provides a framework to describe telecommunications business processes and how these processes use the OSS systems in a typical SP.
- Technology Neutral Architecture: The architecture does not define how to implement NGOSS, rather what principles must be applied for a particular technology specific architecture to be NGOSS compliant. The Technology Neutral Architecture defines architectural principles to guide OSS developers to create OSS components that operate successfully in a distributed environment.
- Contract Interfaces: define an ‘API’ for interfacing NGOSS components to each other across the architecture.

In NGOSS, the TMF has potentially defined an integration architecture which will greatly simplify the process of building large scale OSS systems out of well defined components.

NGOSS Adoption
Unfortunately, while the industry has participated in defining NGOSS through the Telemanagement Forum, the standard has not yet been widely adopted into operational use. Because service providers are not yet specifying the use of NGOSS in new projects, ISV’s have been slow to adapt their current products to the NGOSS architecture.

In a previous paper [3] we described how we felt that this market gridlock could only be resolved by the creation of proof points which would educate the industry as to the value of NGOSS and give confidence to both project architects and product managers to create systems conformant to NGOSS principles. We also described how we believed that to be

---

1 Although in the open source community the mnemonic OSS often refers to Open Source Software, to avoid confusion in this paper we will follow the usual Telecommunications practice of using the mnemonic OSS to mean Operations Support Systems.
successful these proof points would require a significant open source component.

3. OPEN SOURCE IN THE TELECOMMUNICATIONS INDUSTRY

Figure 1 illustrates our view of the status of open source in the Telecommunications industry which has historically been very conservative and slow to adopt the open source model. Until 10 years ago all of the OSS infrastructure would have been either built using large in-house development teams or more recently bought in as Custom-Off-The-Shelf (COTS) products.

![Figure 1 Status of Open Source in Telecommunications Operations Support](image)

The emergence of the internet and the ISP market place drove the commercial adoption of open source infrastructure components such as Linux and Apache. This has forced the telecommunications service provider community to at least evaluate the open source option for infrastructure components. More recently, various groups have been proposing an ‘open source like’ approach to sharing the development of common adaptors between established COTS components. This has been championed by the OSS through Java Initiative (OSS/J) [4] which was established in 2000 by Sun and a number of OSS vendors. OSS/J has sought to embody NGOSS principles and provide a technology specific implementation of the technology neutral NGOSS architecture. It is the first recognised route by which OSS vendors could migrate their components to NGOSS compliance. In addition to the standards work, open source communities such as Tigerstripe [5] are emerging for the tooling and sharing of OSS/J interfaces.

Today there are no viable service provider grade open source OSS components. However we believe these could emerge over the next ten years through a variety of routes. Firstly, service providers could decide to open source internally developed OSS systems and we know of a number of SP’s who are considering this option. Secondly commercial players could enter the open source OSS market. Blue Hat would be one of the first players in this market –although they have yet to release any product [6]. Thirdly, existing enterprise grade open source network management products could mature to the point where they provide an acceptable carrier grade solution. Finally, some of the existing COTS OSS players could decide to open source some part of their existing portfolio.

Since we have established our project, we have encountered a great deal of ‘fear, uncertainty and doubt’ within the Telecos OSS community around open source. This comes in two forms. Firstly, there are those who fear market confusion if we are perceived as intending to provide an open source alternative to existing products. One COTS OSS vendor has jokingly referred to us as the ‘open source terrorists’ (a moniker which we would rather not carry if we are travelling to North America!). Secondly, there are genuine fears in a highly litigious industry about potential patient infringements, anti trust suits and the risk of ‘viral licences’ infecting a production code stream.

It has therefore been very important for us to clearly scope our project around the contribution it is intending to make to the adoption of NGOSS – and not as a potential competitor to the existing OSS vendor community.

4. THE OPEN OSS PROJECT

OpenOSS objectives

Members of the Telemangement Forum have realised that while the management of next generation networks and services poses significant technical challenges, the present supply chain, market configuration, and business practices of the OSS community are an obstacle to rapid innovation and adoption of NGOSS. The present processes undertaken to test, develop and realise Telemangement Forum standards utilise only a small portion of the Telemangement Forum community, which can cause lengthy lead times. Moreover the validation, testing and development of Telemangement Forum standards by implementation are primarily performed within short term ‘catalyst’ projects that discontinue upon completion, or are performed internally within companies where the results are not widely visible to the community. Forums for open development could potentially provide a medium to shorten this supply chain for the deployment of workable systems.

By delivering an open OSS test bed this project hopes to provide a new and complementary route to expedite the testing, development and realisation of Telemangement Forum standards. The software in the test bed will be made permanently available for use by vendors, integrators and service providers, and software will evolve using an open source approach under a license or accreditation scheme to be agreed with the Telemangement Forum technical council.

It is not the intention to create a set of open source OSS at carrier grade standard, but to provide Telemangement Forum members with free access to a permanent and evolvable set of NGOSS reference software that can be extended as the industry chooses.

It is envisaged that this open source approach in the TMF context will help:

- Validate and provide feedback on standards through development and testing of open OSS software based on open source or free software components
- Expedite realisation of NGOSS by service providers through development and sharing of NGOSS compliant adaptors to commercial and open source OSS products
- Promote standards development by providing a reusable pool of “zero-cost”, working NGOSS reference software, infrastructure and tools that can be used in further development projects

The aims of the OpenOSS project are more fully described in the OpenOSS white paper published through the Telemangement Forum [7]

OpenOSS strategy

Figure 2 illustrates the strategy for OpenOSS.
The Telemanagement Forum is in the process of defining a Telecoms OSS Application Map which defines how existing OSS products might map onto the NGOSS architecture. This is designed to help existing OSS vendors to think through what interfaces they will need to make their OSS products NGOSS compliant.

In order to investigate the value of NGOSS, we wish to create a subset of representative components which can fit into the NGOSS application map but which are much simpler than a complete COTS product. This will allow us to think through the integration issues of an NGOSS architecture before the COTS components become available.

These simple components would be constructed using enterprise network management products and use open source NGOSS adaptors which we will create using OSS/J. The value of this approach is that at any point in time, a real COTS component could be substituted into the architecture for solution testing.

In addition we wish to identify a set of open source service components (VoIP Call servers, media servers etc) to allow us to create a realistic network to be managed.

It is important to realise that this is primarily an integration project where we wish to establish good relations with existing open source communities in order to use their projects as the basis for our investigations. Where appropriate we will contribute back to these projects – but we have no desire to establish our own open source platforms where these already exist elsewhere.

Figure 2 Telecoms OSS Application Map and OpenOSS

5. THE OPEN OSS CATALYST

OpenOSS marketing to attract a developer team

Typically to succeed, an open source project requires:

- A critical mass of developers
- Well understood requirements
- Strong involvement from the user community

Although Operations Support Systems are a niche market, it is estimated that world wide over 200,000 IT professionals are employed by service provider IT departments, many of whom are working on internally developed solutions [8]. It is quite possible then that a number of these professionals would value involvement in a project like OpenOSS and would offer an understanding of requirements and close contact with their local user communities. Therefore a key criteria in launching our program was to find a way to attract interest from across this community.

In our industry one of the most effective ways to reach the OSS community is through the Telemanagement Forum. Within the Telemanagement Forum, members can initiate short term Catalyst projects which allow vendors to demonstrate to the membership their experiences of implementation of NGOSS standards. Catalysts last for about 9 months and the results are demonstrated during one of the half yearly Telemanagement World conferences through a rolling catalyst presentation schedule.

Given the aims of the OpenOSS project, we considered it would require the same attention to marketing and business analysis as any conventional product investment strategy. However rather than trying to sell a product, we would be seeking to sell an idea in order to gain active support from across the industry. We identified in [3] that over the long term, the OpenOSS program would need to consider offering a marketing mix as illustrated in Table 1.

Table 1: A marketing mix for open OSS project

<table>
<thead>
<tr>
<th>Mix</th>
<th>Evaluators question</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Price</td>
<td>Does open development reduce the cost through solution life?</td>
<td>Trial projects must provide reduced cost ownership proof points.</td>
</tr>
<tr>
<td>Promotion</td>
<td>Does open source provide a differentiating market channel?</td>
<td>Marketing to ensure high industry visibility (use co-marketing etc).</td>
</tr>
<tr>
<td>Place</td>
<td>Where can the solution be obtained?</td>
<td>Need a well known place (web site) to find the work.</td>
</tr>
<tr>
<td>People</td>
<td>Are a credible development team working on the solution?</td>
<td>Need a governance model which links respected industry players.</td>
</tr>
<tr>
<td>Physical</td>
<td>Process Open development Process /Culture.</td>
<td>Plan how a viable community can be developed around project.</td>
</tr>
<tr>
<td>Evidence</td>
<td>Physical Goods (Product) Can the packaged solution be easily obtained and used?</td>
<td>Packaging/distribution (web download) easing solution use. Initial release to be well documented and fit for purpose.</td>
</tr>
<tr>
<td>Services</td>
<td>Are there sufficient services for the solution?</td>
<td>Need support mechanisms – web logs, FAQ’s and ultimately a commercial support offering.</td>
</tr>
</tbody>
</table>

Table 1 outlines the marketing mix for a fully operational Open Source project. However for the initial proof of concept, we decided to concentrate only on attracting interest through addressing Promotion, Place and People. The Catalyst has provided us with a means to promote the project. The list of sponsors have shown that credible people are involved in the initial stages of the project (see the acknowledgments at the end of this paper) and our initial web site has provided a place to post our work so far [9].

We recognised that going forwards we will have to address much more rigorously the other aspects of the marketing mix including business case proof points, our services strategy and the detailed open source development processes. However for the initial proof of concept, we thought it sufficient to allow...
interested parties to download useful material, and can see that we are progressing plans to take the project further.

6. OPENOSS CATALYST CONTENT

Figure 3 illustrates the content of the Catalyst demonstrated at Telemangement world. The solution was hosted on ten servers at the University of Southampton and accessed from the Telemangement World Catalyst booth using a remote VPN. Five public presentations of this technology were provided every day to a mixed audience of developers and business decision makers.

The development code was shared through a site hosted on Sourceforge. The Southampton and Agilent teams used the Eclipse IDE (with Jboss IDE Plug-in’s) and the Budapest team used Netbeans.

Network and Service layer

In order to provide a realistic Network Management scenario, we constructed a Voice over IP (VoIP) network using a set of Cisco routers and open source VoIP service components. SIP Express Router (SER) [11] is a production quality open source VoIP switch which runs under Linux. SIP Express Media Server (SEMS) [12] is an open source media server which we used to play out recorded announcements. Although open source, both of these components are in use in a selection of service providers to enable their VoIP offerings. We used X-lite, a free to download windows package for our VoIP phones [13].

In order to introduce controlled errors into the network we used NISTNET [14] which is an open source network emulator. This can reduce effective bandwidth, drop or duplicate packets between two addresses according to a routing table which can be set up remotely. In addition we used another open source package, Distributed Internet Traffic Generator (D-ITG) [15] to introduce controlled bursts of overload traffic into the scenario. Thus we proved that it was possible to implement a realistic service network for the purposes of testing a management scenario using open source or free components.

Data Collection Layer

In the data collection layer we used two products to gather data from the network.

On one hand, we used a COTS product, the Agilent NgN Analysis System[16] to monitor the VoIP SIP signalling and tell if the users were experiencing any problems with the VoIP network. Agilent implemented an OSS/J interface on their product so that it could pass data to the Correlation and Presentation layers. The use of a COTS product proved the value of the OpenOSS proof of concept to assist commercial product designers in gathering requirements for integrating their product into an NGOSS architecture.

In addition we used an open source SNMP network management system OpenNMS [17] to monitor the state of the routers in the network and also to monitor the state of the Linux servers.

Correlation layer

For the correlation layer we used an open source inference engine written in Java called Drools [17]. As an alternative we could have used the Sandia laboratories JESS system [18], but the licensing of this implementation would not have allowed us to do any more than a simple demonstration. Drools used the JSR96 javax.rules api which means that it could be relatively easily swapped out for a commercial inference engine if required later.

Presentation layer

We used the Jboss Portal Server to allow us to provide a presentation layer using JSR 168 compliant portlets. In addition Agilent implemented their presentation layer using the apache implementation of the Java Server Faces (JSR 127) technology. We used Open Ticket Reporting System (OTRS) [19], an enterprise grade trouble ticket solution to render our trouble tickets.
OSS/J interfaces
In order to quickly create OSS/J interfaces we used a mixture of solutions. The Budapest University team modified the OSS/J Quality Of Service (QoS) Reference Implementation downloadable from the OSS/J site [20] while the University of Southampton and Agilent teams used a new implementation of the OSS/J QoS interface released under an apache licence by Agilent. All of the implementations were demonstrated running on a Jboss application server [21].

7. EXPERIENCES
It is important to understand that from inception the Catalyst project had 3 months to design and integrate the proof of concept. Therefore it was a high pressure project which unlike other Catalyst projects was using open source components with which none of the parties were highly familiar. In future phases of this project, we hope to have increased our familiarity with the open source components available and be able to concentrate more on defining the service scenarios.

In addition to raising the profile of the OpenOSS project, the catalyst provided the following proof points for the project concept.

Firstly, we proved that suitable open source components exist at a level of maturity for us to proceed with this project. Further more the communities surrounding these projects appear to be accessible and helpful. In particular both the OpenNMS and Drools communities provided us with helpful support.

The disparate companies were able to work together using the open source approach. However the management style had to recognise that the resources were being provided as good will - there were no formal contracts binding us to deliverables. This required flexibility in the definition of the solution to match what each party felt they could contribute.

As a team, we felt we had a lack of experience of the open source development cycle – and this is an area we will wish to develop quickly through our engagements with more established open source projects. In particular, we will need to think through our code management process and decide which development tools are most appropriate going forwards.

The experience of building the proof of concept provided an excellent vehicle for beginning to appreciate how OSS/J and NGOSS can be practically applied. We believe that further work will deliver some really useful experience to the industry. We believe that NGOSS provides a very useful framework for understanding an OSS architecture and we hope to utilise it more fully going forwards.

OSS/J itself is functionally rich but lacking in tool support. Since the Catalyst, we have discovered an emerging open source project [22] seeking to address the OSS/J tooling problem and hope to work with them in the future.

Participating in a catalyst provided a very intense and rapid vehicle for getting a lot of work done collaboratively in a short time for a proof of concept. However, to do so, we did cut a lot of corners and would like to take more time about defining the next stages of the project. However the biggest benefit of the Catalyst was the visibility it gave us across the industry – and the potential contacts to build a sustainable open source community going forwards.

8. NEXT STEPS
Having just completed our Catalyst demonstration in May 2005, our first priority will be to document our work and make the code publicly available as an early proof of concept.

We are in the process of formulating plans for a longer term project which will seek to establish an open source community under the auspices of the OpenOSS Initiative. We are actively seeking partnership across the OSS business community and within the Open Source and Academic communities in order to realise the vision of OpenOSS.

9. CONCLUSIONS
The approach of OpenOSS has been welcomed by the Telemanagement Forum as a means to involve more academic partners in their activities and also provide a ‘living lab’ to demonstrate NGOSS principles. The basic principles of this way of working have been tested through the initial ‘Catalyst’ demonstration and in parallel a program of research and development is being developed which the partners are currently discussing converting into a project consortium.

10. ACKNOWLEDGMENTS
We would like to acknowledge the help of the following companies in pursuing this work: Agilent Technologies, AutoMagic, BT, Budapest University of Technology and Economics, Cable & Wireless, Cognizant Technology Solutions, Colt, Covad, Drools, Invocom, NTTGroup, OpenNMS Inc, QinetiQ, SUN Microsystems, University of Southampton, Vodafone.

11. REFERENCES
[16] (2005) Agilent NgN Analysis System


www.otrs.org


[22] (2005) Tiger Stripe Development Community
http://www.tigerstripedev.net/