Sahana – engineering a sustainable ICT solution for disaster management

Digital Earth '06 – Summit on Sustainability
Auckland, New Zealand – August 27-30, 2006

Gavin Treadgold (gt@kestrel.co.nz)
Committee Member, Sahana Project
Director, Kestrel Group Ltd
Member, New Zealand Open Source Society Inc.

Abstract

Sahana is Sinhalese for 'relief'. Sahana rose from the waves of the Boxing Day tsunami in Sri Lanka as a free and open source software (FOSS) solution for managing information before, during, and after a disaster. Commercial disaster management systems are often unsustainable for all but the wealthiest of countries – Sahana was created as a system that can be deployed in a sustainable fashion in any country on the bare minimum of computer hardware and communications. Many of the benefits provided by FOSS contribute directly to sustainability of this solution – increased flexibility, ease of customisation, deployment not restricted by licence agreements, and skills required to support Sahana can increasingly be found in local communities. A core component of any disaster management solution is the geospatial information about the disaster and its impacts on affected populations. The Sahana development community is investigating various options for publishing spatial information that can be used in Digital Earth applications to ease the visualisation of the significant amounts of data that becomes available after a disaster. Restricted access to 'fundamental' spatial information is one of the largest potential roadblocks to the development of sustainable digital earth solutions for disaster management.

Introduction

The tsunami that struck Sri Lanka on December 26, 2004 spawned vast quantities of information and prompted the development of a free and open source software (FOSS) solution called Sahana – the Sinhalese word for 'relief'.

Within a couple of days, the need for such a system increased as hundreds of Non-Governmental Organisations (NGO) responding to hundreds of thousands of displaced people required co-ordination.

Sahana development was initially led by the Lanka Software Foundation[1] and supported by volunteers from the Sri Lankan IT industry – more than 80 volunteers produced the first implementation of Sahana Phase 1 in 3-4 weeks. Most NGO and donor organisations were registered in Sahana, and over 26,000 families were tracked in the People Registry during this period.

As the immediate need for Sahana subsided in the months following the tsunami, more international contributors became involved in the project. These ranged from programmers wanting to help out, to those that wanted to offer assistance – drawing upon their disaster experiences, including emergency managers.

Longer-term, the goal is to use Sahana as a means of encouraging communities to better prepare for and manage their own events. This will start with providing tools to incorporate plans and reference material such as communication directories in advance, and other techniques to encourage greater inter-agency co-ordination before an event.

The paper aims to introduce the reader to some of the issues surrounding the Sahana Project in terms of building and sustaining an application to assist in the management of disasters. When developing a software solution, there are some fundamental design considerations that are becoming increasingly relevant to interconnected and interoperable applications. The key issues are access to source code, and the availability of public data. Whilst these considerations are mostly philosophical, they have significant implications to the long-term sustainability of the application.

Specifically, Sahana will focus on providing a more sustainable disaster management solution on two levels:

- Application - free and open source software
- Data – open and interoperable standards for data storage, communications and discovery

Additionally, there are project management issues in terms of ensuring that the Sahana Project is itself sustainable.

The Sahana Project

The definition of *sustain* from the Oxford American Dictionary lends itself well to community needs during disaster management.

1. *Strengthen or support physically or mentally*
    - *Cause to continue or be prolonged for an extended period*
    - *Bear without breaking or falling*
2. *Undergo or suffer (something unpleasant, esp. an injury)*

The Sahana Project is designed to sustain communities that have been impacted by significant events, and enable them to operate more effectively over extended periods in times of crisis and suffering, until resilience returns the community to the 'new' norm.

An international community maintains Sahana and all contributions are provided back to that community at no cost – a share-and-share-alike ethos to ensure that everyone benefits. Sahana is free to use and has no licensing costs associated with it.

[1] http://www.opensource.lk/

Sahana has been designed to operate in a diverse range of environments due to the nature of disasters. It can run on web servers, laptops and has even been installed on PDAs. Over time, Sahana will support both standalone and networked modes of operation, and allow communication between multiple Sahana servers, including synchronisation of data.

Sahana is capable of running on a number of operating system platforms including many varieties of Unix including Linux, BSD, Mac OS X, as well as Microsoft Windows operating systems. Currently the core requirements are an operating system that has the following services installed:

- Apache HTTP Server[2] from The Apache Software Foundation[3]
- PHP[4] scripting language by The PHP Group
- MySQL[5] database server by MySQL AB.

All of these are FOSS services. Other server applications are being considered for support, in particular PostGIS[6] with the intention to provide the PostgreSQL database with geographic extensions. This will enable an integrated approach to storing spatial and non-spatial Sahana data, in addition to better integration with a number of FOSS GIS software packages. Currently MySQL does not provide comprehensive support for spatial data.

The version of Sahana (Phase 1) deployed to assist Sri Lanka's response to the tsunami contained the following key modules:

- People Registry – register, track and match victims of a disaster.
- Organisation Registry – register, connect and track NGO's involved in response.
- Camp Management System – register and track camps.
- Request/Assistance Management System – record, track and match requests and offers of assistance.

Several additional modules are planned or currently under development:

- Alerts/messaging – a system of sending alert messages and receiving notifications via mobile and other devices.
- Child protection system – a system designed to track children's needs to enable more effective management.
- Data import/export – a tool to enable the import/export of Sahana data to other formats and applications.
- Disaster impact assessment – a system to capture information about the damage and other impacts of the event.
- Intelligence/information – a system for logging and tracking communication and intelligence related to the disaster.

---

[2] http://httpd.apache.org/

[3] http://www.apache.org/

[4] http://www.php.net/

[5] http://www.mysql.com/

[6] http://postgis.refractions.net/

- Inventory/supply chain/logistics – a system to track and create a market for critical supplies in the affected area.
- Response/rescue team management – a system to register, track and management rescue and response teams in the affected area.
- Volunteer co-ordination – a system to enable management and co-ordination of the volunteer resource, particularly spontaneous volunteers.

The capabilities of Sahana are set to expand with the identification of additional key technologies to provide greater functionality:

- Mapping/GIS, and GPS integration – Sahana can already use Google Maps
    - OpenGIS Consortium – CAT, WFS, WMS, SFS
- Biometrics
- Interoperability/provision of information via OASIS[7] Emergency Management Technical Committee[8]
    - Common Alerting Protocol (CAP v1.1)
    - Emergency Data Exchange Language (EDXL v1.0)
    - OpenDocument Format for Office Applications (ODF v1.0)
- Support of existing paper-based forms
- PDA applications for remote field work
- Operating modes – standalone, networked (LAN) and Internet

Deployments

Sahana has seen official deployments in multiple events including the Sri Lankan response to the tsunami in 2004, the 2005 earthquake in Pakistan, and the Philippines mudslide in 2006. It has also recently seen unofficial deployment in support of the Yogjakartra earthquake and in preparation for an eruption of Mt Merapi. Sri Lanka's largest non-governmental organisation, Sarvodaya[9], is also deploying Sahana within their disaster unit. The majority of the deployments to date have used Sahana Phase 1 (and variants) as used in Sri Lanka.

Sahana and the Digital Earth concept

The geospatial framework for Sahana is currently in development following months of discussions. Although not confirmed as yet, the Sahana approach is likely to draw heavily upon by FOSS geospatial applications and libraries, as well as OpenGIS Consortium data standards.

The spatial components of the Sahana database will be made available using OpenGIS standards such as the OGC Web Mapping Service (WMS) and Web Feature Service (WFS). Discovery of available Sahana geospatial services will occur through the OGC Catalogue Service (CAT) standard.

---

[7] http://www.oasis-open.org/

[8] http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=emergency

[9] http://www.sarvodaya.org/

There are two key elements to geospatial information used in Sahana.

1. Sahana information – the information that is recorded in the Sahana database such as the locations of people, resources and facilities, and impact assessment information. This is the record of the disaster and contains information specific to the event, and in most cases is linked to non-spatial/attribute information.
2. Fundamental layers – these are the mapping layers over which Sahana information can be overlaid.

It is intended that both the fundamental layers and Sahana information will be able to be displayed and navigated in digital earth applications such as NASA World Wind[10]. Initial efforts will be for display-only purposes, but over time the development of a World Wind plug-in that provides read/write access to authorised personnel could allow all management of the disaster information from within the digital earth application. The ability to extend World Wind in this fashion is a benefit obtained because it is a FOSS application.

There are a number of issues associated with the traditional networked worldview applied to the use of digital earth applications. Generally, these require a broadband Internet connection to access remote mapping layers. Clearly broadband may not be available in an area that has suffered significant infrastructure damage from an earthquake. From the initial planning it has always been the intention of the Sahana Project to encourage the deployment of Sahana in standalone mode or self-reliant networks until such time as power and communications infrastructure is repaired.

The key issue is that all fundamental layers must be capable of being downloaded from the supplier and utilised in a network environment disconnected from the Internet. Whilst it is possible to aggregate these fundamental layers after a disaster, more effective response will come from having a process to ensure that the fundamental layers are aggregated and maintained in a map server in advance of an event.

Sustainable Solutions

There are a variety of areas where benefits can be obtained to produce a sustainable solution to disaster management. The key elements that need to be considered include:

- Application – access to source code, development philosophy, and licensing
- Data – storage, interoperability and discovery
- Project management – governance, structures, people, relationships, funding, engagement and communication

The Application - Free and Open Source Software

There are two broad approaches to the development of software, each with a number of variants.

Proprietary software is more easily understood, as it is currently the prevalent model for the development and deployment of software applications. Some of the key features that define proprietary software include:

[10] http://worldwind.arc.nasa.gov/

- No/restricted access to source code
- Single commercial vendor
- Requires purchase
- Licence generally restricts usage and deployment
- Difficult to customise

Traditionally, proprietary vendors have driven software development, and the end user is provided with a software package that generally cannot be directly modified to suit organisational needs. It may be able to be customised using options and configuration settings, but if the software doesn't implement a function it can be difficult to have additional functionality implemented.

FOSS changes that model, as organisations have access to the source code – the human readable code that tells the program how to operate. This is perhaps the most fundamental aspect of FOSS and most other benefits flow from it – access to the source code, the freedom to modify it, and the freedom to redistribute your modifications. Anyone is therefore able to read, copy and modify the source code. This allows anyone to improve software, adapt it, and fix bugs. Additionally, this can often happen at a greater speed than conventional software development.

The Internet has catalysed the spread of FOSS, providing a means of linking programmers and users around the world. The individuals come together on a project-by-project basis to develop software fit for a specific purpose.

FOSS provides a significant number of benefits when compared to proprietary software solutions. A number of these will now be discussed – note that many of these benefits are interconnected, and where possible relationships have been highlighted.

The rights granted by FOSS create the potential for far more sustainable software solutions as detailed below.

Source code access

Access to the source code of an application is perhaps the most fundamental benefit of FOSS applications, and from where many of the other benefits stem. It cannot be understated how important having the ability to not only access, but also to modify and compile source code is.

Low barrier to entry

Having the freedom to download, install, experiment with, and *access the source code* of a FOSS application creates an extremely *low barrier to entry*. This means that the potential exists to involve a significant number of people, and at a minimum create awareness of Sahana as a FOSS solution for disaster management. Compare this to proprietary disaster management solutions for which it is generally not possible to even access a demonstration service on vendor websites without booking a demonstration or receiving a formal presentation from the vendor. More significantly you will not be able to download and experiment with the proprietary solution on your own organisations systems and experiment with specific issues such as integrating Sahana with existing systems.

The low barrier to entry also allows anyone, including individuals, with an interest to become involved in FOSS projects – such as volunteers. It is not necessary to be a government or NGO to contribute.

Increased evolution

It is possible for solutions developed using the FOSS philosophy to benefit from an increased rate of evolution and development. Access to the source code allows any suitably skilled programmer to first build, and if the modification is desirable, share it with the community, and incorporate it into the solution. Even if an organisation does not have experienced programmers, it is possible to hire a programmer to produce a module that produces the required functionality. This is because Sahana is written in a widely available and utilised programming language.

Vendor control of source code does not generally allow third parties to incorporate improvements in proprietary software. This generally results in a slower rate of evolution of proprietary software – evidence of this can be seen in the closing gaps between proprietary and FOSS solutions.

The network effect

The value of a network generally increases with the number of nodes that are members of the network. In disaster management, the larger a network becomes, the easier it will be to share information and better co-ordinate response.

The use of proprietary software can limit the effective size of a network – primary because of the financial cost associated with purchasing the application from a commercial vendor. The free, up-front cost of FOSS applications translates to a relatively *low barrier to entry* for open solutions. There may be other costs that are common to both solutions such as support, training and maintenance that reduce the differential between proprietary and FOSS solutions. If organisations are prepared to invest time in understanding the FOSS solution, FOSS applications can have a far lower cost of entry to a network. Consequently, a network built on FOSS applications and open standards has the potential to scale to a far larger network.

Transparency

The lack of access to the source code of proprietary software means that users have to treat a closed application as a 'black box' where the process of converting an input to an output is not transparent or well understood. The importance of having access to the source code for important societal information systems is becoming increasingly evident – such as having transparency in electronic voting systems in the United States. In disaster management it is similarly important to have transparency in software process, especially as increasingly advanced and automated logic is applied to complex information systems. FOSS solutions are able to provide this transparency as by their very nature *access to the source code* of the application is implied.

Leveraging previous work

Given *access to the source code* of other applications and software libraries, it is possible to 'stand on the shoulders of giants' by incorporating tools that already exist rather than having to build tools from scratch. As it is often not always possible to add FOSS tools and libraries to proprietary software – this is again a benefit primarily enjoyed by FOSS solutions. Proprietary systems generally require additional commercial

tools and libraries, or to task developer time to build them – regardless additional funding and/or developers are required. The corollary is that FOSS projects can do more with less.

In providing map services for Sahana there will, for example, be issues associated with reprojecting geospatial data so that all map layers are correctly aligned (recognising that there may be additional errors that cause the map layers to be incorrectly aligned). Generally, reprojection of geospatial data involves complex mathematical calculations. There exists, however, a set of tools called Proj.4[11] – Cartographic Projection Library – a FOSS solution that provides all of the functionality required. It should be noted that Proj.4 is released under a licence that allows it to also be incorporated in proprietary software. Some libraries and tools are released under more restrictive licences that may not be able to be utilised by proprietary software.

This means that a FOSS project is not required to build these specialist tools, and can better utilise project developers on core Sahana development.

Integration with existing systems

The right to *access source code* means that a FOSS solution can easily be customised to allow Sahana to connect to pre-existing information systems within an organisation. Creating this capability before an event can allow information to be easily imported or linked to from other information system. With a skilled team of developers, it is possible to build such integration after a disaster has occurred.

As most proprietary systems do not allow the end user access to the source code, it can be difficult or impossible to link the system to pre-existing information systems. In some cases the vendor may offer this service, but as they are the sole holder of the source code, costs associated with providing additional integration can be far from competitive, hence unsustainable over the long term.

Local economy benefits

Perhaps one of the key sustainability benefits of FOSS solutions is not technical; but has a basis in economics.

Many existing proprietary solutions for disaster management are created and sold from developed countries with relatively strong global currencies. This creates the situation where either less developed countries are unable to afford the proprietary solutions, or if they do decide to purchase them, then it is probable that a significant amount of the revenue from the sale of the proprietary solution will be repatriated to the country of production. In addition, the weaker local currency, and the requirement for the foreign vendor to generate profits in a relatively stronger currency can cause high prices in the local economy. Naturally, high prices act as a barrier to entry to the market. Finally, vendors marketing their proprietary product in numerous countries may be unable to provide sufficient localisation of the solution for the local economy.

Alternatively, FOSS solutions provide local FOSS vendors a *low barrier to entry* and *access to the source code*. As the local FOSS vendor gains more knowledge and experience with the FOSS solution – they are able to provide in-country training, support, maintenance, and development services, and the revenue and profits from this service remain in the local economy. This strengthens both the local economy, and

[11] http://proj.maptools.org/

expands the range of organisations and people that are able to contribute to, and maintain Sahana. The benefit is a more sustainable approach both at the local and global levels, while ensuring that a suitable level of customisation for the local market occurs.

Shared development

The share-and-share-alike philosophy means that development costs of a FOSS solution can be spread across a potentially large number of organisations. Alternatively, if an individual organisation desires a particular module constructed, they could choose to fund development of just that module, and then contribute the module to the community.

The actions of local FOSS developers can also create global benefits for the applications that they develop – sharing of local development allows global benefits to accrue.

Proprietary software does not generally allow shared development, as there are issues with access to intellectual property and licensing. Proprietary software vendors best protect themselves by not providing other developers with access to the source code, or if they do provide it, then the third party is unable to customise it.

The Data – Discovery, Interoperability and Storage

FOSS provides considerable benefits for software, especially that used in the public interest. There are, however, significant issues around data that must be incorporated into creating a sustainable approach to disaster management information systems.

The information systems that organisations use for routine operations are typically designed to suit the internal needs of the organisation and are unlikely to have the capability to interoperate with other organisations information systems.

In the period leading up to, during, and after a disaster, a great deal of community co-ordination is required. Critical to response and recovery co-ordination is the ability to discover sources of information, to communicate the information using recognised protocols, and to store it in accessible formats.

Community response to disasters can place significantly different demands on organisations routine information systems. Due to the customised nature of most organisations' routine information systems, the organisations are generally not well positioned to immediately share information other than via formal interoperability standards such as email, and informal standards such as word processing documents and spreadsheets. Using simplistic mechanisms such as these can result in significant inefficiencies and a less effective response.

The review by the Government Flood Review Team into the February 2004 flood event found a number of deficiencies in interoperability and the sharing of geospatial information. The following recommendations were made in the Reid Report.

- Development of a single, consolidated mapping GIS
- Development of systems that support single data entry
- Few examples of agencies sharing and integrating data
- Lack of awareness of databases and spatial information available

Interoperability Benefits

There are a number of significant benefits that can be obtained from adopting open standards for discovery, storage and interoperability.

- Access to data in proprietary formats or databases – web services can provide an interoperable public interface to data that is stored in a (possibly proprietary) database or formats.
- Utilise data using the most appropriate tool – data stored in widely recognised formats can often be opened in a wide variety of applications, allowing the user to choose the most suitable tool for the task at hand. Data stored in a proprietary format can generally only be used in the application it was designed for, and may limit functionality.
- Data 'suffers' network effects – as more data is stored in an open and interoperable format, that format will benefit from network effects similar to applications. It will encourage more organisations to provide information in that standard, and more libraries, tools and applications will be developed to support the standard.
- Notification and aggregation of content – interoperability standards allow the monitoring of a wide range of different systems, with little consideration given to their underlying technical architectures. They also allow easy aggregation of information from these systems.
- Discovery of publicly accessible data – it can be difficult to manually search and discover available public data sources. Interoperability standards support the easy publishing of available sources of data to directory or catalogue servers that make discovery of the resource significantly easier to find.

Standards in Sahana

The ability to share information between multiple Sahana servers and other applications is going to play a key role in developing a solution that can be easily integrated into organisations existing systems and applications. The key approach to achieve this is through interoperability standards for data exchange.

The delivery of Hypertext Markup Language (HTML) over the Hypertext Transfer Protocol (HTTP) represents the minimum level of interoperability that Sahana provides – in that it will be accessible from any modern web browser. As the functionality provided by Sahana increases, the number and complexity of standards required also increase, as well as the means of accessing the information.

There are a number of standards that the Sahana Project is looking to utilise as a means of improving data interoperability (current version at time of writing shown):

- OASIS OpenDocument[12] v1.0 second edition (ODF) – XML formats used to store word processing, spreadsheet, charts/graphics and presentation documents.
- OGC Geography Markup Language[13] v3.1.1 (GML) – a format designed to transfer geospatial information from servers to clients in XML. Primarily a format

[12] http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=office

[13] http://www.opengeospatial.org/standards/gml

for transmission of geospatial data, it could easily be used as a storage format as well.

- OGC Web Feature Service[14] (WFS) – an XML interface for allowing requests of geospatial information across networks, the 'payload' within WFS is GML.
- OGC Web Map Service[15] (WMS) – an XMl interface for allowing requests of geospatial information across networks, with the results being returned in raster and vector files.
- OGC Catalogue Service (CAT) – supports the ability to publish and search collections of descriptive information (metadata) for data, services, and related information.
- OGC Simple Feature Access 2 SQL[16] v1.1.0 (SFS)
- GPS Exchange Format[17] v1.1 (GPX) – a lightweight XML format designed for the interchange of Global Positioning System data (waypoints, routes, and tracks).
- OASIS Common Alerting Protocol v1.1[18] (CAP) – an XML-based data format for exchanging emergency alerts and public warnings.
- OASIS Emergency Data Exchange Language[19] v1.0 (EDXL) Distribution Element – a suite of emergency data message types including resource queries and requests, situation status, message routing instructions needed in response.
- People Finder Interchange Format[20] v1.1 (PFIF) – a data model and an XML-based exchange format for sharing data about people who are missing or displaced by natural or human-made disasters.
- Really Simple Syndication[21] v2.0 (RSS) – a standard for web content syndication.
- RFC 4287 The Atom Syndication Format[22] (ATOM) – a standard for web content syndication.
- Geo Really Simple Syndication[23] (GeoRSS) – a means of georeferencing syndicated web content, also works with ATOM.

The Project – Maintaining the Momentum

An effort was made in August 2003 to start a project to develop a FOSS solution for disaster management entitled OpenEOC[24] (Open Source Virtual Emergency Operations

[14] http://portal.opengeospatial.org/files/?artifact_id=8339

[15] http://portal.opengeospatial.org/files/?artifact_id=14416

[16] http://www.opengeospatial.org/standards/sfs

[17] http://www.topografix.com/gpx.asp

[18] http://www.oasis-open.org/committees/download.php/14759/emergency-CAPv1.1.pdf

[19] http://www.oasis-open.org/committees/download.php/17227/EDXL-DE_Spec_v1.0.html

[20] http://zesty.ca/pfif/

[21] http://www.rssboard.org/rss-specification

[22] http://tools.ietf.org/html/rfc4287

[23] http://www.georss.org/

[24] http://osveoc.sourceforge.net/

Centre). OpenEOC was unsuccessful in producing a single line of code. There was a well-recognised need for open source disaster management software, but there were no real drivers to progress development of a solution at that time.

In contrast, the initial version of Sahana was developed on the fly responding to very real and life-threatening needs. Little opportunity existed to spend time researching and planning potential solutions, rather a solution had to be constructed immediately. This resulted in the production of Sahana Phase 1 that was used for the recovery from the tsunami. As news of this project spread across the Internet, additional volunteers expressed interest in contributing to this project.

Domain experience

The programmers that contributed to Sahana Phase 1 worked under difficult circumstances having little planning or prior knowledge of issues associated with the management of disasters.

The lack of awareness of broader, non-technical issues associated with disaster management meant that before the tsunami, the developers would not have been the best placed to develop an application for disaster management. However, during the development of Sahana and the response to the tsunami, they gained a lot of hands-on awareness of issues associated with disaster management – very valuable experience that is difficult to come by. The lack of formal training meant their experience was limited solely to the event(s) that they were involved in.

This presents a risk to the project in terms of being able to provide strategic guidance to developers based on expert knowledge of the domain of disaster management. This obstacle was addressed by involving individuals that had years of experience in humanitarian assistance, emergency management or the management of large technical projects.

Volunteer direction

There are some risks associated with FOSS projects that are based entirely on the contribution of volunteers – the most notable being that it can be difficult to direct volunteers to work on a specific aspect of the application.

Volunteers in FOSS projects will tend to focus on areas that hold specific interest to them, for example incorporating a particular technology into an application. This results in certain aspects of development, including what could be considered “boring, yet essential” elements of the project to receive less development. If these elements are not developed in a timely manner, especially if they represent a core component of Sahana, it can reduce the momentum of a project and risk the long-term sustainability of the project.

Mitigation of the project risk has taken the form of funding a team of core developers to contribute to the project. Funding gives an element of control and direction to the development process that is not guaranteed with volunteer developers. This statement is not designed to understate the commitment and service provided by volunteer developers, rather to identify a failure in the process to appropriately direct the developer resource.

Following the tsunami, funding was provided by the Swedish International Development Cooperation Program (SIDA) to ensure that internships could be provided through the Lanka Software Foundation for the first year (August 2005 – July 2006) of development

of Sahana Phase 2. This allowed a core team of developers to be directed by the Sahana Project Manager.

Communication and awareness

Volunteer projects can often suffer from ineffectual efforts at communication and marketing due to a lack of time. This can result in a project losing momentum by not obtaining enough interest through a lack of awareness in the project.

Raising the project profile attracts support for, and provides leverage for contributors to obtain organisational endorsement of their work. In addition it provides visible recognition for contributors.

Strategic management

There are some challenges associated with getting professional people involved in FOSS projects – especially at the strategic level. The biggest challenge can be obtaining enough volunteer hours out of professionals that already have limited hours to commit – this is a common problem with any form of volunteerism in developed countries.

A hybrid approach may be required to be able to pay an honorarium or hourly pay to allow professionals to commit to a FOSS project like Sahana. This ensures that professionals from appropriate disciplines are able to guarantee their commitment to the project. This is similar to the approach taken with core developers, in that a financial commitment needs to be made to the core strategic leaders of the project to ensure that they are able to provide consistent strategic management for the Sahana Project.

Key Challenges and Possible Solutions

There are a number of steps to overcoming the challenges of creating a sustainable ICT solution for disaster management.

1. FOSS is not well understood
    a. Researching and understanding FOSS and its benefits relative to commercial solutions
    b. Seeking to apply it where information needs to be shared between different systems.
2. Limited access to fundamental public data – it is difficult to find and access a significant amount of publicly funded data.
    a. Greater efforts need to be made to release information funded by tax/rate-payers.
    b. Information released need to be accessible under permissive licences so that the data can be downloaded and deployed on systems not connected to the Internet.
3. Lack of funding – required to ensure project momentum and development of Sahana is maintained.
    a. Look for local and global funding opportunities to ensure a continued stream of funding is available to maintain project momentum and ensure that core technologies and fundamental modules continue to be developed.

b. Commit funds to core development and strategic management teams.

4. Failure to engage – engagement is required to ensure the project delivers appropriate solutions to disaster management problems, and the people and communities that need them.
    a. Create awareness of the project.
    b. Engage organisations to become involved in the design, development, testing and deployment of Sahana.

Conclusion

Sahana is one of the most positive outcomes from the tsunami. It is unfortunate, but the need created by the tsunami generated sufficient momentum to produce a free and open source solution for disaster management, something that other projects have not yet been able to achieve.

In the same way that the Ballantynes fire hastened the development of fire safety in New Zealand, the Sahana Project needs to build on the efforts started in Sri Lanka following the tsunami to build a sustainable ICT solution for disaster management.