OSBR.CA

The Open Source Business Resource

JULY
2008

JULY 2008

PUBLISHER:
The Open Source Business Resource is a monthly publication of the Talent First Network. Archives are available at the website: http://www.osbr.ca

EDITOR:
Dru Lavigne
dru@osbr.ca

ISSN:
1913-6102

ADVERTISING:
Rowland Few
rowland@osbr.ca

GRAPHICS:
Ryan May

Much has changed since the term "open source" was coined ten years ago (http://opensource.org/history). The idea of sharing code, once relegated to research institutions and hobbyists, is starting to reach ubiquity in the consciousness of both government and business. More recently, this concept has extended beyond software. Everything from hardware to data to knowledge is becoming "open". Despite this trend towards openness--or perhaps due to the stark contrast openness provides--accessibility remains a difficult obstacle to overcome.

The issues surrounding accessibility take many forms. In the first article, Pierre-Paul Lemyre from the University of Montreal reminds us that everything old is new again. While the difficulties surrounding the open source licensing of software are still being resolved, those same difficulties are now being experienced when providing open access to data and knowledge. He proposes a global licensing repository as a solution towards providing access while complying with licensing terms. Marco Zehe from the Gnome Accessibility Project concentrates on ensuring that the features found in software applications are accessible to all, including those with disabilities.

Ensuring proper governance is another piece of the accessibility puzzle. After all, how can a business take advantage of open access if there aren't policies in place to govern proper usage? Andrew Back from Osmosoft proposes that governance, employee education, and tools to monitor compliance are needed for effective business operations. In this month's TIM Lecture Series, Michael Weiss from Carleton University discusses innovation through mashup ecosystems and Mahshad Koohgoli from Protecode discusses the importance of creating a bill of materials to manage software components.

In this month's question and answer section, Glenn McKnight, an open source consultant, answers the question "Besides compliance with legislation or standards which govern Internet accessibility, are there any business reasons for maintaining an accessible website?". We also have recent reports on open source business models, the challenges in providing open access, and a guide for open access authors.

As always, we look forward to your feedback.

Dru Lavigne

Editor-in-Chief

dru@osbr.ca

Dru Lavigne is a technical writer and IT consultant who has been active with open source communities since the mid-1990s. She writes regularly for O'Reilly and DNSStuff.com and is author of the books BSD Hacks and The Best of FreeBSD Basics.

"Letting your data connect to other people's data is [...] not about giving to people data which they don't have a right to. It is about letting it be connected to data from peer sites. It is about letting it be joined to data from other applications. It is about getting excited about connections, rather than nervous."

Tim Berners-Lee

The success story of open source software (OSS) makes us see very clearly that in a networked world, centralized production of information is not the only viable model. It is now largely understood that distributed production can often equal and surpass it, both in quality and quantity. This has led people in all disciplines to rethink their relationship with information, giving birth to a plethora of initiatives generating value by promoting the mass collaboration of individuals over shared sets of information. Based on rich Internet applications, wikis, social tagging or social networking technologies, these initiatives gave birth to a revolution that has been dubbed Web 2.0. Whether they originate in non-profit or business ventures, they all add up to the ever increasing mass of accessible and reusable information.

For current information hubs that have been developed through independent channels, it is anticipated that the next step in the evolution of the web will make seamless integration possible. This development should create tremendous opportunities for those capable of building innovative services and knowledge products on top of this shared knowledge base. In fact, along with the technological foundations of this web of ideas, practical commercial implementations are already starting to appear. However, these early experiments highlight the fact that the most important challenge to overcome might not reside in the technology itself. Instead, the management of rights may, more than anything else, hinder the efficient aggregation of distributed information.

Accessibility, Reusability, and Interoperability

Software developers realized a long time ago that while access to information is one thing, the ability to reuse it is another. For them, binary code and restrictive software licences stood as a solid barrier between the two concepts. In response, some chose to adopt an alternative development model promoting the sharing of source code with permissive software licences. While some of these licences, like the Berkeley Software Distribution (BSD) licence, simply favour a certain level of reciprocity among developers, others, such as the GNU General Public License (GPL), go further and secure the openness of the code they cover.

More recently, the open source approach to licensing has been expanded to cover a wider range of contexts. This resulted in the apparition of hybrid development models offering both the possibility for users to adapt software to their respective needs and the preservation of some restrictions on its circulation. Altogether, these experiences have shown that software is a much more valuable asset when it is reusable. In addition, they have demonstrated that diverse reuse conditions can fit diverse needs and expectations.

To some extent, a similar evolution occurred for other kinds of information circulating over the Internet. Following in the footsteps of OSS developers, innovative entrepreneurs have learned to adapt and expand collaborative development models to produce a large array of information products.

This resulted in the creation of impressive information commons such as Wikipedia (http://www.wikipedia.org), media repositories like Flickr (http://flickr.com) and Youtube (http://www.youtube.com), social bookmarking systems like del.icio.us (http://del.icio.us) and Digg (http://digg.com), as well as social networking websites such as Facebook (http://www.facebook.com) or LinkedIn (http://www.linkedin.com).

Once again, the development of dedicated licensing schemes has been crucial in this outcome. The Creative Commons (CC, http://www.creativecommons.org) movement, in particular, has been extremely helpful in clarifying the spectrum of rights and reuse conditions that can be attached to shared information. This, in turn, has led to domain specific licences, such as the Australian Free for Education licence (http://www.aesharenet.com.au/FfE/) which allows the free circulation of information for education purposes while imposing conditions on other types of reuse. The experiences of the last few years have shown how businesses can thrive on accessible information by promoting different forms of reusability.

As a consequence, the volume of information accessible on the Internet under technical and legal conditions that make reuses possible is growing at an incredible pace. Up to very recently, the collaborative initiatives driving this transformation have evolved independently from each other. While pictures uploaded by Flickr users are distributed under permissive terms, their reuse is still mostly limited to other users of the same service. The same can be said about all of the flagship initiatives of the Web 2.0 revolution. Because the information flow of these web services has been limited to the vertical direction, most of the accessible data is now compartmented into separate information silos.

While silo construction might have been inevitable, the possibility to reuse the information they contain calls for improving the reciprocity between sources of information. For this goal to be achieved, a horizontal information flow must complement the current one. Data from one source must become mixable with data from other sources through various layers of services. Interoperability it is said, is the key to this puzzle.

The World Wide Web Consortium (W3C, http://www.w3.org) has tackled this issue for several years. Thanks to its efforts, technical solutions making interoperability possible are widely available and documented. They include data modeling languages like XML and RDF, syndication technologies such as RSS and ontology (http://en.wikipedia.org/wiki/Ontology_%28information_science%29) standards like OWL. The W3C hopes to encourage web developers to annotate the information they disseminate, creating a computer readable web parallel with the current human readable one. Doing so would give a comprehensible meaning to data, allowing its dynamic discovery, rearrangement and execution. This is what has come to be known as the web of ideas, or Semantic Web, which was recently renamed Giant Global Graph (http://dig.csail.mit.edu/breadcrumbs/node/215) by Tim Berners-Lee. Unfortunately, if the idea of a fully semantic web is appealing in theory, its practical implementation has mostly been limited to the academic field.

The approach taken by industry has been slightly different. Using heuristic, or text-recognition technologies, businesses aiming to leverage semantics to gather accessible information from around the web have started to appear. Focusing on partial sets of data from specific fields, they manage to recognize limited ranges of concepts and to associate data from distinct sources accordingly.

Over the last couple of years, this form of limited semantics have given birth to practical applications. The first wave revolved around specialized search engines, such as Spock (http://www.spock.com). The second wave seems to be oriented toward shortcuts, or the analysis of content to quickly deliver additional information. Yahoo! Shortcuts (http://shortcuts.yahoo.com) and Lingospot (http://lingospot.com) stand out as promising initiatives in this category. Using artificial intelligence to automatically create links between distributed data, all of these services are partially circumventing the requirement for technical interoperability.

Whether they will ultimately take one form or the other, the role and scope of semantic technologies are bound to increase over the next few years. Under their influence, it can be envisioned that the aggregation and integration of the important volume of information that is already accessible and reusable will soon become technically possible. Undoubtedly, this outcome could generate completely new markets for information products, taking advantage of everything that flows over the networks.

Entrepreneurs currently involved in the development of semantic technologies are realizing that interoperability has more than one side. In addition to understanding the meaning of the data, they are increasingly confronted with the necessity to understand the legal conditions attached to it. Indeed, the diversity of the restrictions imposed on the reuse of accessible information by the gigantic number of existing copyright notices and licences is the most important obstacle to its aggregation. Because of this challenge, automated reuse of information originating on the web needs to be limited to preselected sources that can be trusted. Otherwise, reproduction of the content must be avoided. As odd as it may seem, the old issue related to the proliferation of licences is coming back to haunt the next generation technology.

The Fragmentation of Rights

Web content licences, just like software licences, are found in an ever increasing number of forms for the simple reason that copyright holders are free to control the reproduction of their works as they see fit. The wide range of diverging motivations, commercial interests, and business strategies has made the fragmentation of rights according to infinite reuse conditions inevitable. The specifics of the various formats under which online content can be distributed as well as the existence of distinct domains of application have also contributed to the phenomenon.

Moreover, the need to adapt licences to the context of various jurisdictions that often have conflicting legal requirements has created an additional layer of complexity over the licensing landscape. In the end, the difficulty to manage the resulting diversity of possible terms and conditions is amplified by the fragmentation of rights down to the smallest elements of information. Reuse restrictions are not necessarily attached to entire websites, or even to specific web pages. Distinct licences can potentially govern every bit of data they disseminate.

The difficulties generated by this situation are not fundamental as long as humans are fully in charge of the reuse of information. However, the efficient aggregation and integration of distributed information and the successful implementation of a semantic web require computers to manage this process, at least partially.

To achieve this, they first need the capacity to retrieve the applicable licences for the available information. Second, they require a mechanism to resolve the actual meaning of these licences. Third, they must be capable of selecting only the information disseminated under adequate conditions for the anticipated reuse. If any of these three operations proves impossible to automate, it is probable that the recent innovations in the field of semantic technologies will never reach their full potential.

If the possibility to aggregate reusable information from the web once again puts forward the problem of the fragmentation of rights, that problem has been addressed by several organizations in the past. The Free Software Foundation (http://www.fsf.org) has repeatedly warned OSS developers against the threat posed by the proliferation of licences to the compatibility of source code. Specialized products such as those developed by Black Duck Software (http://www.blackducksoftware.com) are specifically designed to address this issue.

The CC has promoted the most effective measure against proliferation through its set of standardized licences. By encouraging web developers to embed licensing information into their content, CC eases its retrieval by crawlers and other web robots. By providing a computer readable version for each licence, it makes the resolution of their terms possible. By standardizing terminology and keeping the number low, it also facilitates any posterior selection to be made by third parties. For all of these reasons, content distributed under CC licences has been central to aggregation efforts undertaken up to now.

Although the numerous merits of the CC approach cannot be challenged, it does not entirely solve the issue generated by the fragmentation of rights. While a growth in web content covered by its licences certainly increases the volume of information becoming available for aggregation, it does little to deal with the mass of reusable data that is not (and often cannot be) distributed under a CC licence. The fact that copyright holders have the right to attach alternative restrictions to the circulation of their works, coupled with the understandable policy of CC not to automatically accept any new licence proposal, accounts for the need to develop a more encompassing solution. The best illustration of the limitations resulting from this situation is probably the Google search engine feature entitled "usage rights" (http://www.google.com/advanced_search?hl=en). Because it entirely relies on CC tagging of web pages, it completely ignores all of the text Wikipedia made available under the terms of the GNU Free Documentation License. It is precisely to fix this problem that a higher level resolution mechanism is required.

A Global Licences Repository?

Can the CC vision of a lawyer readable, human readable, and computer readable version of copyright-related information be expanded to all licences covering content circulating over the Internet? The legal code of relevant licences being accessible online and its standardization being out of the question, the only workable solution might lie in the conception and implementation of a database of licences and their respective conditions. Organizing the multitude of licences under a single template would allow for the streamlining of their resolution and selection. It would also allow for the development of a web service that could be queried indifferently by users and computers.

Obviously, there are a large number of obstacles that may prevent the completion of such a repository. The large number of reuse conditions, as well as the numerous format and domain specific restrictions, are certainly barriers. Issues related to the internationalization and the versioning of licences are another. In addition, managing compatibilities between licences in order to make relicensing possible can prove to be a daunting task. Nevertheless, paths can be imagined to circumvent each of these obstacles. Conditions and restrictions could be organized into groups or categories. Licences could be managed at the lowest possible level and related ones associated together. The designation of compatibility could be limited to the most common licences.

Notwithstanding its design, the successful implementation of a global licences repository would also depend on the proper interaction of several key elements. The large-scale adoption of a tagging model allowing the effective detection of licensing information by content aggregators is one of the most important. The involvement of a community of users in feeding and updating the database is another as central management would be impossible to achieve.

The nature of the data involved calls for the necessity to generate trust in the system by insuring transparency and adequate quality control procedures. Finally, extended use of the repository will only occur if its outputs are provided in a large range of standardized formats matching the various requirements of extremely diverse users, in conjunction with simple communication tools facilitating the interactions with the system.

Conclusion

Although this proposal would have sounded like an extremely ambitious undertaking only a few years ago, OSS and other collaborative initiatives have demonstrated successes in distributing and managing efforts adequately. Automating the management of licensing information will require substantial investments of knowledge and energy by a broad range of players. Ultimately, it will need to be done for the web to reach its next phase of evolution. Otherwise, the fragmentation of rights will continue to impede technologies allowing the dynamic discovery of data from ever achieving their promise of opening the large-scale reuse of distributed information.

Pierre-Paul Lemyre is one of the leading researchers of LexUM (http://www.lexum.umontreal.ca/index_en.html), the legal information technologies laboratory of the Law Faculty at the Université de Montréal. He is highly interested in the improvement of access to legal information, the challenges that lasting development poses, as well as in the legal issues related to free and open source software.

"The power of the Web is in its universality. Access by everyone regardless of disability is an essential aspect."

Tim Berners-Lee

While it seems intuitive that applications, especially open source ones, should allow all to use and enjoy them, many developers are unaware of the need for accessible applications. Providing accessibility in information technology is not difficult, but it does require a basic understanding of different types of disabilities, commonly used assistive technologies, and the special accessibility features built into languages and standards. Most of all, accessibility requires a conscious effort and a desire to include everyone.

The accessibility of computer software has seen drastic improvements over the past two decades. This article reviews this progress, examining the technologies developed and offering guidelines for developers to create accessible applications.

The Need for Accessibility

Up until recently, the largest driving force behind desktop computing environments has been Microsoft, first with MS DOS, followed by variants of Microsoft Windows. These operating systems were not designed with the needs of people with disabilities in mind. Many, including those who were blind or physically disabled, were unable to use applications which were written for Microsoft operating systems. These applications assumed that computer users could:

- read and react to text and images displayed on the screen

- type on a standard keyboard

- select text, pictures, and other information using a mouse

- react to sounds played

The last point is somewhat less of a limitation as most software doesn't rely exclusively on audio to relay feedback.

If a person was unable to do one of the above-listed tasks, they found themselves unable to use many popular computer applications. Here are some of the groups of people that have problems doing some of those tasks:

1. Print disabled: blind, low vision, obstructed vision, dyslexic, cognitively disabled and illiterate individuals.

2. Physically disabled: users with amputations, paralysis, repetitive stress injury, cerebral palsy, muscular dystrophy, Parkinson's or other problems limiting mobility.

3. Hearing impaired.

We must also consider the increasing numbers of aging baby boomers that are beginning to experience problems with their vision or dexterity. When you add all these groups together, a lot of potential users emerge.

Alternative Ways to Access Screen Contents

Most computer programs are so visual, they are difficult or impossible to use for persons with visual impairments. This need not be the case. Here's how nonprint readers use desktop software today:

1. Text-to-speech (TTS): Those who can't read print usually use talking programs. TTS is also useful for other print disabilities such as dyslexia and for those who cannot speak, in place of their own voice. Finally, this technology could be useful to mainstream users either on portable information appliances or to access information when the eyes are busy elsewhere.

2. Magnification: enlarges the screen's contents. For those with low vision, it may suffice to use a larger font, a built-in high contrast theme, or an extra large screen. Otherwise, screen magnification programs may be used. These allow zooming in to portions of the screen, while following the mouse or the current focus. Screen magnifiers also have some built-in TTS and the ability to filter text and images through various color palettes, such as black on yellow for high contrast, or green on blue for low contrast.

3. The Optacon (http://en.wikipedia.org/wiki/Optacon): provided access to printed words, graphics and on-screen information by means of an array of vibrating pins the size of an index finger. The user used one hand to read the vibrating pins, and the other hand moved a mini-camera over the material to be read. Unfortunately, the unit is not currently produced, although there is occasional talk of resurrecting this useful device.

4. Braille: a solution used for quiet reading, for detailed work, and by deaf-blind users. This can come in the form of hard copy braille printed on braille embossers, or from a refreshable braille display. These technologies require special drivers, braille formatting routines, and software based text-to-braille translation. The importance of braille itself must be emphasized. For those that read it, braille can offer higher levels of employment and life fulfillment.

Audio and braille based user interfaces are concepts for which software designers are not historically trained. Dealing with information when you're blind is like seeing everything through a mail slot: sequentially and methodically. Only small pieces of sequential, non-graphical information can be conveyed via TTS or a refreshable braille display. Whatever the user does, the software needs to respond with small, bite sized pieces of information. Ideally, intelligent decisions are made by the software so the user does not have to wade through non-relevant data.

Alternative Ways to Enter Data

Another problem is how people with disabilities get information into the computer. If you're physically disabled, you may not be able to type on a regular keyboard or use a mouse. Here are some of the alternative ways physically disabled people enter information:

1. Sticky keys: make entering key combinations easy. For example to make a capital letter, first press the shift key, release it, then press the letter to be capitalized. The sticky key technique is utilized by people who have only one usable hand, or who have no use of their hands and type using a stick in their mouth.

2. Single switch: these technologies enable persons with severe physical disabilities. Some users enter information by choosing from lists of options. They might press a switch down to begin moving a highlight bar through the list, and release the switch when the desired option is highlighted.

3. Special keyboards: exist to make data entry easier. However, any special features are generally handled in the keyboard itself so that no special programming is required.

4. Speech recognition: allows people to talk to the computer. This technology has come a long way, but still needs to be more integrated into mainstream software.

5. Consistent keyboard support and hotkeys.

Testing with people that have disabilities generally benefits everyone. Use the accessible toolkit checklist to make sure your user interface (UI) controls adhere to standards (http://www.mozilla.org/access/toolkit-checklist).

The Lack of Context

To meet the needs of disabled users, many accessibility hardware and software vendors create products and software to help people who can not perform one of the aforementioned four basic tasks. Some examples of these assistive devices and software include: i) screen reading software; ii) TTS; iii) alternate input devices; iv) voice recognition software; v) screen magnification software; and vi) comprehension software, which allows a dyslexic or learning disabled computer user to see and hear text as it is manipulated on the computer screen.

An entire adaptive technology industry has grown to meet the accessibility needs of disabled users. One place to learn more about this industry is the CSUN (http://www.csun.edu/cod/) conference in Los Angeles, which takes place every year in March.

The solutions developed by accessibility vendors have greatly increased the employment and personal fulfillment opportunities of hundreds of thousands of persons with disabilities, and the importance of their work cannot be diminished.

However, these solutions fall short of providing people with disabilities with a working environment which is completely accessible and usable. This is due to a simple problem of context: a user's interaction with a computer is governed by the situation in which this interaction takes place.

When the user types something on the keyboard, or when an application displays text or images on the screen, the exact meaning of these actions is determined by the context in which they take place. For example, one application might display the image of a light bulb to indicate that it is processing a task, while another might display it as an indicator that it has completed processing a task. Without the application somehow notifying a blind person about the meaning of each of these light bulb images, the blind person is unable to understand what the application is attempting to convey. Similarly, voice recognition software often needs information about the context of a user's interaction, in order to make sense out of what the user is speaking. This context problem still plagues modern accessibility aids and solutions.

The first notable attempt at solving this problem was put forth by Microsoft in 1997, and is called Microsoft Active Accessibility (MSAA, http://msdn.microsoft.com/en-us/library/ms697707.aspx). This initiative realizes that complete accessibility is not possible without cooperation between applications and accessibility aids such as screen reading software or voice recognition software. The MSAA defines a Windows-based standard by which applications can communicate context and other pertinent information to accessibility aids.

This solution has seen only partial success, largely due to the fact that it requires significant changes to the applications being made accessible. Because most popular desktop and productivity applications are not open source, this forces disabled people to rely on the companies which produce the software to make it accessible. These companies are often reluctant for various reasons, including the large amount of time required to modify the original application.

On a positive note, recent federal purchasing rules, such as Section 508 (http://en.wikipedia.org/wiki/Section_508_Amendment_to_the_Rehabilitation_Act_of_1973), have caused many companies to pay attention and implement MSAA support.

Enter Open Source Software

Microsoft was on the right track with MSAA, but because the source code to most popular desktop applications used in large corporations is not publicly available, they were never made fully accessible. In open source, however, making the necessary accessibility modifications is very possible.

Open source software (OSS) is an ideal way to meet the needs of disabled users. Accessibility can be fully integrated into the core design, rather than tacked on as an afterthought. OSS also gives disabled programmers a chance to control their own destiny, by giving them the opportunity and the right to directly fix the inaccessible software themselves.

Furthermore, any software solution that can enable equality should by all rights be free of charge. If no special hardware is required, why should a disabled person pay extra money to use the same software as everyone else? That said, there is still an important role for adaptive technology vendors in creating special services and hardware, or even proprietary software on platforms where that is appropriate. The ideal situation would be for adaptive technology professionals to make money in the underserved areas of rehabilitation, training and support. Each end user has a unique set of problems, and in the open source world, providing highly customized solutions can be a business in itself.

A number of companies have set out to improve on MSAA and further develop accessibility application programming interfaces (APIs) that would benefit everyone. Under the umbrella of the Linux Foundation (http://www.linuxfoundation.org/), a group from IBM, Mozilla, Sun Microsystems and several assistive technology vendors have developed an enhancement to MSAA called IAccessible2 (http://www.linuxfoundation.org/en/Accessibility/IAccessible2). IAccessible2 is fully compatible with MSAA and enhances accessibility in areas where MSAA has weaknesses. With IAccessible2, access to rich content such as web pages, word processing or spreadsheet documents, or multimedia presentations is possible without having to rely on screen analysis techniques for context. This guarantees much more accurate access to rich content, allowing:

- both screen reading and screen magnification software to present a better picture to the visually impaired user
- voice dictation software to more accurately interface with such applications to implement features such as "say and select"
- alternative input devices and software to interface with all possible elements without having to rely on screen coordinates or other such inaccurate mechanisms

Lessons were learned from earlier tries at making the Linux graphical user interfaces more accessible using the GNOME accessibility toolkit (ATK) APIs and the Gnopernicus screen reader and magnifier. This knowledge was transformed into the GNOME Assistive Technology Service Provider Interface (AT-SPI, http://developer.gnome.org/projects/gap/). This interface allows for a range of open source assistive technology solutions available for the GNOME desktop:

- the Orca (http://live.gnome.org/Orca) screen reader, for people with visual disabilities, offering speech and braille output and magnification functionality

- the GNOME on-screen keyboard (GOK, http://www.gok.ca/) software from the University of Toronto's Adaptive Technology Resource Centre

Another software project to take advantage of these improved accessibility APIs on Linux is Jambu (http://www.oatsoft.org/trac/jambu), which supports alternative input for motor-impaired computer users who are able to only use one switch.

In parallel, several open source projects were enhanced to support the AT-SPI APIs. Most of the software included in the GNOME desktop such as Gedit, Pidgin, and Terminal, as well as many mainstream projects such as Mozilla Firefox, OpenOffice.org/StarOffice, Rhythmbox, or Tomboy are more accessible today than they were a few years ago. There are several success stories from people who actually make a living or got a new job because of the support of accessible applications on Linux.

Developers' Guidelines

As a developer, there are several suggestions for making your applications accessible. At the Mozilla Project, we encourage developers to follow the general front-end accessibility requirements. In particular, there are a number of potential gotchas when developing with the Mozilla XUL (http://en.wikipedia.org/wiki/Xul) UI. Developers should follow the practical techniques listed in the Accessible XUL Authoring Guidelines (http://developer.mozilla.org/en/docs/Accessible_XUL_Authoring_Guidelines). These guidelines cover many scenarios. If you take time to learn them, they will become an unconscious improvement to your design and engineering technique.

Ensuring correct keyboard accessibility when developing new controls is important for providing consistency. Mozilla's XUL and HTML widgets (http://en.wikipedia.org/wiki/Widget_toolkit) already support proper keyboard accessibility.

New controls should support MSAA/IAccessible2 and ATK via the cross-platform nsIAccessible (http://developer.mozilla.org/en/docs/nsIAccessible) interface. Engineers can provide context simply by creating an nsIAccessible for each custom control and the infrastructure to do this is straightforward.

Conclusion

No matter what kind of work you do, the basis of accessibility is in the understanding that every user is different. The exact techniques may change depending on the engineering environment. There are many resources available to application developers for creating accessible applications, several of which are mentioned in the Recommended Resources section at the end of this article.

This article is based upon the Mozilla document "Software Accessibility - Where Are We Today?". The original is available from the Mozilla website (http://www.mozilla.org/access/today).

Marco Zehe works as quality assurance engineer for accessibility at Mozilla Corporation. His responsibility is the accessibility of the Firefox web browser and Mozilla platform. His responsibilities include making sure that all web and user interface content is properly exposed to assistive technologies on all supported platforms, helping to test new enhancements such as IAccessible2, and to drive adoption of standards forward within and around Mozilla. Before joining Mozilla, Marco worked as a second-level support and localization manager at Freedom Scientific and has been in the assistive technology industry since 1996.

Recommended Resources

Mozilla Accessibility Project
http://www.mozilla.org/access/

Links and Resources in Accessibility
http://www.mozilla.org/access/resources

Marco's Accessibility Blog
http://www.marcozehe.de/

Dive Into Accessibility
http://diveintoaccessibility.org/

"...that experience - of a CIO not knowing how ubiquitous and valuable free software has become to their organization - isn't atypical. In fact, it's the norm, and a divide we're gently trying to bridge. Opportunity's everywhere. So is free and open software. They might even travel in pairs."
Jonathan Schwartz, CEO of Sun
(http://blogs.sun.com/jonathan/date/20080414)

The case for the effective operation of Free/Libre and Open Source Software (F/LOSS) in the enterprise has never been stronger. Yet in some quarters, the chasm between senior management's perception of the penetration of F/LOSS within their organization, and the reality, has never been wider. And when you consider that Gartner predicts that “by 2012 more than 90% of enterprises will use open source in direct or embedded form” (http://www.gartner.com/DisplayDocument?id=638643), this suggests that the development of an effective F/LOSS policy will become increasingly necessary for business operations.

Many enterprises, for now, are sourcing the majority of their F/LOSS solutions via a vendor. This does not remove the need for governance. Even with commercial arrangements in place, it is crucial that business have an understanding of F/LOSS communities: what drives them, how to interact with them, and what obligations they may have to them. The game has changed and innovation is no longer the reserve of software vendors with large development budgets. Software development is now enabled by open licenses that afford great freedoms and, in doing so, facilitate widespread collaboration. With this unprecedented pace of innovation, comes new obligations. We argue that the need for education around F/LOSS communities and licensing is clear.

In this article we discuss the areas of governance, education, and tools which together constitute the base capabilities required for the effective enterprise adoption of F/LOSS technology and principles.

Governance + Education + Tools = Operations

The management of F/LOSS in the enterprise starts with governance and the creation of a policy and process to support the effective and appropriate adoption of associated technology and principles. This will, at the very least, amount to a general policy on the usage of F/LOSS, but will also likely require developer and procurement specific policies. If the standard desktop software policy is relatively strict in terms of what can and cannot be downloaded and installed, the developer policy will need to account for this while still enabling innovation and ensuring continued network integrity. The procurement policy and supply contracts can require that suppliers disclose use of F/LOSS in integrated solutions as well as provide all the materials that are required to meet licensing obligations.

Once the new policy and process is in place, it is important to provide a widespread awareness and common understanding not only amongst developers, but across development managers, product managers, research, procurement and anyone else who may come into contact with F/LOSS. Without this, the business may be exposed to risks that vary from lost opportunities to innovate, up-skill, and reduce costs, to the potential for litigation. Education is a key enabler for the effective enterprise adoption of F/LOSS and its value cannot be overemphasized.

Language

In naming the mandated organizational unit, F/LOSS Operations or Open Source "Operations" is preferable to F/LOSS or Open Source "Governance". This supports the wider role of being an enabler, rather than simply being responsible for policing and restricting the use of F/LOSS.

It is easy to adopt the negative language of F/LOSS detractors and those looking to sell associated tools and services, and this should be avoided. For example “licensing issues” and “risks” may be used in connection with license obligations and support arrangements. Here, F/LOSS is just like any other software in that you must: i) have a license to use it; and ii) either be able to support it yourself or have a support contract with a third party. These are not new considerations. But since F/LOSS enables new possibilities in both areas, it is likely that enterprises will need to bring their understanding up to date in order to build confidence in F/LOSS adoption.

Reuse

When new needs have been identified, organizations should look to reuse existing tools and capabilities where appropriate. Functions engaged in the management of F/LOSS – for instance, in an operations or governance capacity - should not seek to replace, subvert or undermine existing agencies but to instead work in concert with them. F/LOSS should largely be treated just like any other software that has the same considerations around architectural conformance, security, support and the right to use. Mutually beneficial relationships can be formed with the organization's IT architects, procurement, security officers, and legal counsel.

In fact, this is critical in effecting outcomes that are in the best interests of the business and internal stakeholders, while meeting or exceeding any obligations to external communities.

Obligations

Meeting licensing obligations to external development communities is of the utmost importance, as F/LOSS is no different to proprietary software in that you must have the right to use it. If you do not abide by the terms in the license, you have no right to use the software. If you disagree with those terms, it follows that you'll either have to look elsewhere or speak to the developer(s) and see if they are willing to provide the software on different terms via a paid-for license. Here, governance is required in support of projects looking to work with F/LOSS.

More often than not it is likely that, given a license and your intended use case, your obligations will be clear. However, there will be instances where license compatibility needs to be examined, such as when combining software from multiple sources for redistribution. In such cases, legal guidance may be needed. The role of an operations/governance unit could be considered as triage to your legal counsel. However, where there is any doubt, no matter how small, consult your lawyers.

Increasingly, companies are beginning to appreciate that the maximum value from F/LOSS can only be realized through a symbiotic relationship with the community. In many cases, keeping modifications made to F/LOSS closed, rather than protecting investment, can lead to increased support costs, reduced ability to innovate, and missed opportunities. As such, a developer policy may need to cater for employees contributing to existing F/LOSS projects and starting new projects.

Here there is an obligation to the business to ensure that all the relevant stakeholders have provided sign-off and accept the licensing model proposed for a given development.

Where a company has a patent portfolio, additional checks may be required to ensure that furnishing code via a F/LOSS license will not compromise an existing patent. It is also possible that standard employment contracts may state that the company owns all work done by an employee. Therefore, Human Resources personnel may need to be involved in the review of the relevant company policy.

Tools

When considering new tools required by the enterprise, it should be noted that F/LOSS equivalents may be tricky to find, select, and implement. Vendors will offer their own solutions, and these should be considered alongside solutions produced by the F/LOSS community. For example, the source code analysis tool FOSSology (http://www.fossology.org/) is freely available from the FOSSbazaar (https://fossbazaar.org/) initiative, a working group of the Linux Foundation. For some, FOSSology will suffice as a tool for identifying license types in enterprise code bases, thus enabling them to ascertain obligations. Where the supply chain is more complex and there is concern that fragments of unattributed F/LOSS code may have been integrated along the way, the more fully featured capabilities of a proprietary alternative may be more appropriate.

Community

When evaluating commercial services, consider that, in some cases, a great deal more value may be had from an investment in education and community engagement.

Much can be gained from a symbiotic relationship with a F/LOSS community, some of which may be unforeseen. A direct relationship with the community provides access to core developers, opportunities to up-skill in-house staff, and possible opportunities to steer development of the project. Such access to core development functions and an opportunity to influence was previously, with proprietary software, largely the reserve of Fortune 500 companies.

Contributing back to a project may actually reduce costs. As an example, contributing patches or enhancements avoids the costs of operating a self-supported forked code base. Contributions also win favour with the community and visibly demonstrate to a global community your organization's capabilities. With a vendor sandwiched in-between, however, a great many of these opportunities are lost, as they exploit the relationships in each direction purely to their benefit.

Conclusion

Even if you think you don't use F/LOSS, chances are that you will consume it indirectly as part of integrated and hybrid solutions. At its most basic, F/LOSS governance will be required in the form of a clear policy and simple lightweight supporting process. You may additionally need to build new capabilities to support effective governance.

Tools from the F/LOSS community should be considered alongside those from proprietary vendors. In the management of F/LOSS--consumption, contribution and creation--consideration should be given to both internal and external stakeholders, and, wherever doubt exists, support sought from domain experts such as your legal counsel.

As an enabler of unprecedented flexibility in terms of innovation, licensing and support, education will be required to build enterprise wide confidence and to avoid confusion whilst addressing misunderstanding. To realise the maximum potential from F/LOSS, you need to understand that it enables you to do what was previously not possible with proprietary software. It is likely that a direct relationship with the F/LOSS community will play a key part. Lastly, it is important to remember that in many respects F/LOSS is no different to proprietary software, and simply affords new optional freedoms.

Andrew Back has been working with free and open source software since 1995. He currently holds the position of Open Source Strategist at BT Osmosoft (http://www.osmosoft.com/). During his tenure at BT he has been responsible for driving strategy for the effective adoption of open source technology and principles across all lines of business, and the creation of BT Design's Open Source Focus Group. He is a Fellow of the Royal Society for the encouragement of the Arts, Manufactures and Commerce, and a Chartered Member of the British Computer Society.

Recommended Resources

Framework for Governance in Open Source Communities
http://opensource.mit.edu/papers/Governance-in-OpenSourceProjects.pdf

FOSS Governance Fundamentals
https://fossbazaar.org/?q=filemanager/active&fid=25

Best Practices in Open Source Governance
https://fossbazaar.org/?q=filemanager/active&fid=17

"The Web was originally designed to be mashed up. The technology is finally growing up and making it possible."

Aaron Boodman, Greasemonkey creator
(http://www.greasespot.net/)

The TIM Lecture Series provides a forum that promotes the exchange of knowledge between university research and technology company executives and entrepreneurs. Readers outside the Ottawa area who are unable to attend the lectures in person are invited to view upcoming lectures in the series either through voice conferencing or webcast. Instructions for joining a lecture are available (http://www.talentfirstnetwork.org/wiki/index.php?title=Instructions_to_join_via_voice_conference_or_webcast).

On June 11, 2008, Michael Weiss from Carleton University delivered a presentation entitled "Open APIs, Mashups and User Innovation". This section provides the key messages from the lecture. Michael's lecture examined the structure of the mashup (http://en.wikipedia.org/wiki/Mashup_%28web_application_hybrid%29) ecosystem as well as implementation issues, including licensing. The slides from the presentation are available for download (http://www.talentfirstnetwork.org/wiki/images/f/f8/Open_APIs_mashups_user_innovation_June_11.pdf).

User Created Value, Open APIs, and Mashups

The first section of the talk focused on the value provided by mashups where the innovation comes from the self-interest of users instead of from the application designers. Successful mashups change the economics of innovation as the value isn't in the applications per se, but in providing contexts for interaction.

Moreover, opening the underlying API (application programming interface) creates unique value as differentiation to under-served users. This differentiation can occur in the long tail (http://en.wikipedia.org/wiki/The_Long_Tail) and allow users to help themselves.

With mashups, the focus is on adding value rather than learning a new language or investing time in coding an application. The value provided needs a clear demarcation point between free and monetized content. The Freemium (http://en.wikipedia.org/wiki/Freemium) model isn't the only, or necessarily the best, monetization model.

Mashup risks include: i) dependence upon API sources; ii) becoming a victim of an open API which isn't open source: when the API disappears, the mashup becomes useless; and iii) if mashups are so easy, how do you differentiate and attract users? While no mashup has been patented yet, there are companies trying to patent mashups (http://www.marengoip.com/patents/20070214430.pdf). Licensing still needs to be ironed out for combining data and it needs to be understood in simple terms. For example, Google's terms of service (http://www.google.com/accounts/TOS) does not allow you to use their APIs with their ads removed. This brings up the issue of API licensing and the fine line between enabling in order to create value while protecting your monetization. There is also a need for research mashups to find research knowledge.

When opening an API, it needs to be simple and well documented. As to support required to release APIs, you need to create well documented tools with constrained functionality in order to reduce usage errors.

Other key messages from this section included:

- client side mashups eliminate the need for n-tier infrastructure

- Jakob's Law (http://www.useit.com/alertbox/20000723.html) implies that instead of spending money on web design, you should spend effort on getting linked everywhere

- the programmable web (http://www.programmableweb.com/apis/directory) has nearly 800 APIs to choose from

Small is the new big. Start small and then attract content. Even Google started small--it was able to leverage its value as it grew.

User Innovation & Mashup Ecosystem

The second half of the lecture provided many practical considerations when working with mashups. Key messages from this section included:

- start with an under-served niche as these are your early adopters

- start with well established APIs

- mashup early and often and see what attracts users

- mapping the mashup ecosystem offers important insights into introducing your own API or mashup

- you can become a victim of your own success if mashup becomes popular; Twitter suffers from this problem of providing reliable infrastructure to handle traffic requirements (http://blog.twitter.com/2008/01/we-had-rough-night-but-now-were-back.html)

While tools are available to visualize APIs and mashups, it is hard to tell whether or not mashups are refactored over time from the data at the programmable web as many refactored mashups are shown as new mashups.

Users should be aware that mashups do not deal with inconsistencies in data. In the future, service level agreements (SLAs) can be used to provide reliable data.

Mashups may be a backlash response to the complexity introduced by web standards. This can also be seen in that Resource Oriented Architecture (http://en.wikipedia.org/wiki/Resource_oriented_architecture) usage is outstripping SOAP (http://en.wikipedia.org/wiki/SOAP).

After geospatial, the next big thing in mashups will probably be social networking. However, maybe the next big thing is not a particular application category, but the use of APIs and mashups as a way of creating extensible products.

Finally, licensing combined data is a yet unsolved problem. Mashups may provide a way to launder data as its original source becomes harder to trace as it goes through various mashups.

Michael Weiss holds a faculty appointment in the Department of Systems and Computer Engineering at Carleton University, and is a member of the Technology Innovation Management program. His research interests include open source ecosystems, services, business process models, social network analysis, and product architecture and design. Michael has published on the evolution of open source communities and licensing of open services.

Recommended Resources

Case Study: Mashups Interoperability and eInnovation
http://cyber.law.harvard.edu/interop/pdfs/interop-mashups.pdf

Why Mashups Matter
ftp://ftp.software.ibm.com/software/lotus/lotusweb/portal/why_mashups_matter.pdf

"Intellectual Property (IP) is one of the least understood and most poorly managed assets of most organizations and may represent either the single largest revenue opportunity - or the most significant drain on profitability."

IPX Whitepaper
(http://www.ipxco.com/pdf/Risk_Management.pdf)

On June 18, 2008, Mahshad Koohgoli, CEO of Protecode, delivered a presentation entitled "Practicing Safe Software: Good Software Record". This section provides the key messages from the lecture. Mashad's lecture discussed the drivers behind establishing software IP (intellectual property) pedigree as well as the preventive and corrective methods of detecting and managing external IP in a project. The slides from the presentation are available for download (http://www.talentfirstnetwork.org/wiki/images/d/dd/Practicing_safe_software_June_18.pdf).

Problem Statement and Issues

Section one of the lecture focused on good record keeping as an essential part of quality software development.

Code contamination is not a problem that is unique to open source software (OSS) as commercial code can also contain IP contamination and can contaminate code. Without records, nobody really knows what is in the software product. Keeping good records on what goes into software is essential; doing it manually is impractical and painful. Large and small companies alike suffer from lack of records and IP uncertainties.

Doubts about IP cleanliness in software have commercial consequences. If there is uncertainty, it is much harder to convince customers to purchase and the company's value decreases in the event of a merger or acquisition (M&A, http://en.wikipedia.org/wiki/M%26A). Very little contamination can result in serious problems. For example, in Veritas vs. Microsoft, (http://www.wapatents.com/VeritasReport%20by%20SepcialMaster.pdf) 56 lines were rewritten from C to C++ with two lines remaining verbatim. Moreover, license compatibility is a key challenge. It is not sufficient to say that component 1 and component 2 have clean IP as the licenses for both components need to be compatible. Enforcement is occurring in the marketplace. German courts aggressively go after GPL violators (http://www.gpl-violations.org/) and the US has seen many high profile cases. However, most cases are still resolved quietly without going public.

IP contamination is still occurring even though software processes have improved. For a long while there were no automated software-content record keeping solutions. There is a need for software governance. You can't leave it to the developers to know what is important to record--policies are important. Don't leave it to the lawyers to fix as this is always too late and too expensive. For developers and even their managers, it is difficult to interpret licensing terms for both commercial and OSS.

Indemnification affects scale. Very large companies assume it as a risk of doing business. Many company's margins are so low, they can't provide indemnity and therefore miss out on some business opportunities.

Know what is in your product--what you don't know can hurt you. Open source philosophy has its merits, and we must still respect the IP of open source creators who choose to retain copyright and to use strict licensing terms. IP has significant value.

Why is this a problem now when software has been around for over 50 years? It is a combination of very competitive business practices, more efficient software processes, and the ease of finding and incorporating external code. Moreover, a lot of the new creativity isn't building from scratch, but from combining existing modules. There is a business opportunity in automated record keeping tools for both lawyers and companies who wish to resolve IP violations.

Risk of IP contamination may be overblown due to the difficulty and cost of finding violations and pursuing litigation. In Canada, you'll be fined for the defendants' costs if you lose the lawsuit.

Solutions

The second half of the presentation discussed the types of IP tracking solutions. Corrective solutions analyze the finished software. With these, you can't detect what you can't identify; that is, the template must be in your database in order to detect it in your code. Therefore, corrective solutions need large and thorough IP databases to work reasonably well.

Record keeping and detecting and creating records of external content should be transparent to developers, unless the company policy dictates otherwise.

Bill of materials (BOM, http://en.wikipedia.org/wiki/Bill_of_materials) provide information about the software components, including information needed to enforce policy and perform due diligence. The software supply chain is anchored around the BOM. Software is usually seen as an art, but it is evolving into a manufacturing process. BOM may also be useful in secure software assurance.

With regards to existing products rather than greenfields, some companies don't want to know what is in current code and instead have a clearly defined point where record keeping begins. Others want to annotate older code.

Is recycled code released as OSS a problem? A company's problem is anything that violates their policy.

Most IP violations are unintentional. An employee can make fraudulent records, but record keeping retains the trail. Many employment contracts contain clauses against plagiarism.

Legal advice is always needed to assess a company's particular IP risk and liabilities. Good record keeping helps the legal process.

Mahshad finished the presentation with a quick tour of his product from Protecode (http://www.protecode.com). Protecode automatically makes records of the content that ends up in software, and can automatically check the content against the policies. Automated software-content record keeping is unobtrusive and transparent to the developer.

Protecode can make records as code is created/imported or traverse and identify an existing software branch in the firm's repository. It allows you to define a record keeping policy by: i) acceptable licenses; ii) minimum size of code to be analyzed; and iii) corresponding actions for violations and unknowns.

Mahshad Koohgoli is CEO of Protecode Inc., a software IP management company. Mahshad has been in the industry for a long time, has a BSc and a PhD from the University of Sussex in England. He holds various patents. He was the founder and CEO of Nimcat Networks, and founder of Spacebridge Networks and Lantern Communications Canada. He held senior roles in Newbridge, Bell Northern Research and Nortel.

Recommended Resources

Ensuring Software IP Cleanliness
http://www.osbr.ca/ojs/index.php/osbr/article/view/463/411

IP Ingredients
http://www.ipingredients.org/

Q. Besides compliance with legislation or standards which govern Internet accessibility, are there any business reasons for maintaining an accessible website?

A. While compliance is generally viewed in negative terms, compliance can provide business opportunities. Before discussing the opportunities, we'll provide a quick overview of the potential market and existing legislation.

Because of the many types of disabilities and the varying degrees of severity, it is difficult to obtain accurate numbers on the number of disabled persons. Estimated statistics include:

- 19.4% of non-institutionalized civilians in the United States (http://www.infouse.com/disabilitydata/disability/)

- 12% of Canadians (http://www.statcan.ca/english/freepub/11-008-XIE/2003004/articles/6804.pdf) and somewhere between 5% and 30% of Canadian children (http://www.ohrc.on.ca/en/resources/discussion_consultation/Education?page=Education-IV_.html)

- the United Nations estimates that there are at least 650 million persons with disabilities worldwide (http://www.un.org/disabilities/default.asp?id=695)

For those with a disability, the scope of the Internet technology (IT) problem goes beyond access to websites to any IT used in the creation, conversion or duplication of data or information. This includes, but is not limited to: fax machines, photocopiers, automated banking machines, telephones, and multimedia.

Legislation regarding equal access does exist. North American examples include:

- Americans with Disabilities Act (http://www.ada.gov/)

- Ontario Human Rights Commission (http://www.ohrc.on.ca/en)

- Treasury Board of Canada Secretariat's Common Look and Feel Standards for the Internet (http://www.tbs-sct.gc.ca/clf2-nsi2/index-eng.asp)

With regards to Internet accessibility standards, the World Wide Web Consortium (W3C) updated their Web Content Accessibility Guidelines 2.0 (WCAG 2.0, http://www.w3.org/TR/WCAG20/) in April, 2008. The United Nations has published The Standard Rules on the Equalization of Opportunities for Persons with Disabilities (http://www.un.org/esa/socdev/enable/dissre00.htm).

Despite the number of disabled persons and existing standards, much of the IT world remains inaccessible. In December, 2006, the United Nations commissioned an audit on Internet accessibility for persons with disabilities (http://www.nomensa.com/resources/research/united-nations-global-audit-of-accessibility.html). Of the 100 websites evaluated during the audit, only three achieved the basic accessibility rating.

The 1999 Understanding the Digital Economy conference noted that "As we examine the civil rights requirements for access and fairness, note the additional benefits that accrue: by mainstreaming the functionality found in the assistive computer arena into the architecture of our digital economy, we will expand technological innovations and develop creative solutions."

"By embracing our individual differences, the collective community receives greater benefits than that achieved through the segregation and isolation of people with disabilities" (http://www.icdri.org/CynthiaW/the_digital_divide.htm). By realizing these untapped markets, business opportunities begin to appear. IT consultants can assess, redesign, build and implement WCAG 2.0 compliant websites and retrofit government kiosks. IT companies can generate niche solutions meeting the market for adaptative technologies. There is an entire market segment that requires affordable devices using open projects such as http://openprosthetics.org and http://www.tackledesign.com.

A robust and strong digital economy requires the removal of barriers through the deployment of accessible design elements on our computers, information technology and communications. By directing our research and policy directives to address these problems, we will overcome the digital divide and ensure full participation in the global digital economy.

Glenn McKnight has worked extensively with Canadian and international private and public sector organizations to promote cost effective IT and non-IT projects. His work included Baygen Radio of South Africa, environmental technologies in China, and IT strategies in India. His certification experience includes developing apprenticeship programs, operating IT schools, and promoting the Linux Professional Institute as an international standard.

Recommended Resources

Accessibility in the Virtual Workplace: Opportunities and Challenges

Industry Canada's Assistive Technology Links
http://www.at-links.gc.ca/zx10000E.asp

Human Resources and Social Development Canada: Disability Issues
http://www.hrsdc.gc.ca/en/disability_issues/index.shtml

Universal Instructional Design
http://www.tss.uoguelph.ca/projects/uid/

Ontario Accessibility Directory
http://www.accessibilitydirectory.ca/

The goal of the Talent First Network Proof of Principle (TFN-POP) is to establish an ecosystem anchored around the commercialization of open source technology developed at academic institutions in Ontario.

The priority areas are the commercialization of open source in:

- Mapping and geospatial applications
- Simulation, modeling, games, and animation
- Conferencing
- Publishing and archiving
- Open educational resources
- Social innovation
- Business intelligence
- Ecosystem management
- Requirements management

Expected Results

The TFN-POP is expected to:

- Establish a healthy ecosystem anchored around the commercialization of open source assets
- Maximize the benefits of the investment in the Talent First Network by the Ministry of Research and Innovation
- Accelerate the growth of businesses in Ontario that use open source assets to compete

Eligibility to Receive Funds

Individuals eligible to receive funds are faculty, staff, and students of universities and colleges in Ontario.

Budget and Size of Grants

A total of $300,000 is available. Applicants' requests should not exceed $30,000.

The TFN-POP may provide up to 50 percent of total project costs.

Criteria

Proposals will be judged against the following five criteria:

- Strength and novelty of open source technology proposed
- Extent of market advantage due to open source
- Project deliverables, likelihood that the proposed activities will lead to deliverable completion on time, and effectiveness of the plan to manage the project
- Track record and potential of applicants
- Extent of support from private sector

Application

The electronic version of the application received by email at the following address: TFNCompetition@sce.carleton.ca will be accepted as the official application. The email must contain three documents: a letter of support, project's vitals, and a project proposal.

Letter of support: (maximum 2 pages) a letter, signed by the person responsible for the Technology Transfer Office or Applied Research Office of the academic institution that proposes to host the project and the faculty member or student who will lead the project, must be included. This letter should describe the nature of the support for the project from the academic institutions, companies and other external organizations.

Project's vitals: (maximum 1 page) The project's vitals must include:

- Person responsible for applied research or technology transfer at the college submitting the proposal: name, mailing address, telephone number, and email address

- Project leader: name, mailing address, telephone number, and email address

- Team members: names, mailing addresses, telephone numbers, and email addresses

- Budget: Total budget, with TFN's contribution and that of other organizations

- TFN investment: TFN contribution broken down by payments to students, payments to faculty, and payments to project awareness activities

Project proposal: (maximum 5 pages) Project proposal must include the following:

- Benefits: (maximum 1/2 page) Description of the benefits of the proposed project, and an overview of the context within which the project is positioned

- Advantage: (1/2 page) Market advantage provided by open source assets used in the project

- Information on applicants: (maximum 1.5 pages) Background information to help assess the track record and potential of the people who are key to the project and the college

- Project plan: (maximum 2.5 pages) Description of the deliverables (what will be delivered and when); key project activities; nature of the involvement from companies, and other external organizations; and plan to manage the project

Evaluation & Deadline

Proposals will undergo review by the Expert Panel established by the TFN-POP. The Chair of the Panel may contact the applicants if required. A final decision will be communicated to the applicants within 30 days after the email with the official application is received.

There is no deadline. Applications will be evaluated on a first-come basis until the $300,000 available is committed.

Contacts

Luc Lalande: Luc_Lalande@carleton.ca

Rowland Few: rfew@sce.carleton.ca

About the Talent First Network

The Talent First Network (TFN) is an Ontario-wide, industry driven initiative launched in July 2006 with the support of the Ministry of Research and Innovation and Carleton University. The objective is to transfer to Ontario companies and Open source communities: (i) Open source technology, (ii) knowledge about competing in Open source environments and (iii) talented university and college students with the skills in the commercialization of Open source assets.

The Bee Keeper: A Description of Professional Open Source Business Models

Copyright: James Dixon

From the Foreward:

The Bee Keeper is a brilliant analogy for the architecture of participation that is at the root of the success of open source. This paper elegantly explains how it can be possible that everybody wins when many contribute rather than pay, and some pay rather than contribute.

http://www.pentaho.org/beekeeper

Open Access – Opportunities and challenges – A handbook

Copyright: European Commission and the German Commission for UNESCO

From the Foreward:

The debate on open access is controversial and complex, with stakeholders displaying widely contrasting opinions. I strongly believe that we must work towards solutions that offer the research community rapid and wide dissemination of results. At the same time, I am convinced that there must be fair remuneration for scientific publishers who invest in tools and mechanisms to organise the flow of information and the peer review system. I welcome this handbook which presents the various views of major stakeholders and covers a wide range of issues relevant to open access. I view it as a very useful and timely contribution to the debate on open access.

http://ec.europa.eu/research/science-society/document_library/pdf_06/
open-access-handbook_en.pdf

Understanding Open Access in the Academic Environment: A Guide For Authors

Copyright: Kylie Pappalardo

From the Preface:

This Guide...aims to provide academic authors with an overview of the concept of and rationale for open access to research outputs and how they may be involved in its implementation and with what effect. In doing so it considers the central role of copyright law and publishing agreements in structuring an open access framework as well as the increasing involvement of funders and academic institutions. The Guide also explains different methods available to authors for making their outputs openly accessible, such as publishing in an open access journal or depositing work into an open access repository.

http://eprints.qut.edu.au/archive/00013935/01/
Microsoft_Word_-_Final_Draft_-_website.pdf

Open Access Authors Fund

June 23, Calgary, AB

University of Calgary professors and graduate students will now have access to a $100,000 Open Access Authors Fund designed to increase the amount of publicly available research. The new fund is the first of its magnitude in Canada. The new fund will provide U of C faculty and graduate students with financial support to cover Open Access author fees.

http://www.ucalgary.ca/news/june2008/authorsfund

Availability of Governance and Intellectual Property Management Software

June 24, Ottawa, ON

Protecode announced the general availability of its software development tool for governance and Intellectual Property (IP) management. The latest release enables commercial software developers and open source creators to accelerate managed adoption of open source code in a simple, painless process. Additionally, the software is now available to the Eclipse community for anyone working on an active Eclipse project. Protecode automatically generates records of software content, identifies and reports associated pedigree and licensing information by checking its properties and compliance against an organization's policies, establishing IP ownership and creating a software Bill of Materials (BOM).

http://biz.yahoo.com/bw/080624/20080624005154.html?.v=1

Eclipse Delivers Its Annual Release Train

June 25, Ottawa, ON

The Eclipse Foundation and the entire Eclipse community are pleased to announce the availability of the Ganymede Release, the annual release train developed by the Eclipse community. The Ganymede Release is a coordinated release of 23 different Eclipse project teams that represents over 18 million lines of code. Seven Eclipse Packages have been created to make it easier for developers to download multiple projects.

http://www.eclipse.org/org/press-release/20080625_ganymede.php

TFN Teams up with The Code Factory to Drive Innovation

June 27, Ottawa, ON

Ontario's Talent First Network and Carleton University's Foundry Program announce a partnership with TheCodeFactory to assist technology savvy students to work, meet and connect with the Ottawa start-up community. The partnership arrangement provides entrepreneurial students and recent graduates access to TheCodeFactory's collaborative co-working space and incubator.

http://www.talentfirstnetwork.org/wiki/index.php?title=CodeFactory-TFN-Foundry

Open Access Policy to Cancer Research Data

June 30, Ontario

The Ontario Institute for Cancer Research (OICR) is taking the lead in 2008 and making the research it funds available to the public through an open access policy that takes effect July 1. OICR's policy, "Access to Research Outputs," provides the guidelines for OICR's scientists when they publish their work and describes the institutional repository where all publications from OICR scientists will be deposited for public accessibility. The policy, which builds on the policy in place at the Canadian Institutes for Health Research (CIHR), requires OICR researchers to provide unrestricted access to their publications within six months of publishing, either through self-archiving of the journal article in the OICR Institutional Repository or through publication in open access journals.

http://www.oicr.on.ca/portalnews/vol2_issue3/access.htm

National Film Board Launches Open Beta

July 3, Canada

The National Film Board is opening its vaults, with 70 years worth of groundbreaking documentaries and cutting-edge animated films being placed online. These films are accessible to all Canadians and free of charge for home viewing. The online repository was written in Python within the Django environment, and uses PyAMF, Apache, Twisted and MySQL. The application is hosted on an Ubuntu Linux server.

http://beta.nfb.ca

August 4-8

Agile 2008

Toronto, ON

Agile 2008 will be an exciting international industry conference that presents the latest techniques, technologies, attitudes and first-hand experience, from both a management and development perspective, for successful Agile software development. Agile 2008 puts attendees in contact with the latest thinking in the agile domain, enriching our collective body of knowledge, influencing the line of thought in the field, encouraging debate, and fostering innovative ideas. It brings together executives, managers, software development practitioners and researchers from labs and academia. The conference is not about a single methodology or approach, but rather provides a forum for the exchange of information regarding all agile development technologies.

http://www.agile2008.org/

August 10-14

74th IFLA General Conference and Council

Quebec, QC

Open Source, distributed services delivery, web services and smart clients provide new paradigms for delivery of library services technology to small and special libraries. The theme of the 74th World Library and Information Congress is "Libraries without borders: navigating toward global understanding".

http://www.ifla.org/IV/ifla74/call-it-en.htm

September 22-26

Toronto Tech Week

Toronto, ON

The second annual TorontoTechWeek will raise national and international awareness of Toronto as one of the largest, most innovative and fastest growing Information & Communications Technology (ICT) markets in North America. A world-class series of events will bring Toronto's ICT community together to promote and foster partnerships, employment, investment, education and business opportunities.

http://www.torontotechweek.com/

September 24-25

How to Manage, Monitor and Measure Social Media Effectively in Your Organization

Calgary, AB (English) & Montreal, QC (French)

Get solutions to your most pressing social media challenges through practical examples and case studies presented by a select group of practitioners who have implemented social media at their organizations.

http://www.canadianinstitute.com/bus_corp_general/SocialMedia.htm?PageMode=Search (Calgary)

https://www.institutcanadien.com/ventes/mediassociaux.htm (Montreal)

The goal of the Open Source Business Resource is to provide quality and insightful content regarding the issues relevant to the development and commercialization of open source assets. We believe the best way to achieve this goal is through the contributions and feedback from experts within the business and open source communities.

OSBR readers are looking for practical ideas they can apply within their own organizations. They also appreciate a thorough exploration of the issues and emerging trends surrounding the business of open source. If you are considering contributing an article, start by asking yourself:

1. Does my research or experience provide any new insights or perspectives?

2. Do I often find myself having to explain this topic when I meet people as they are unaware of its relevance?

3. Do I believe that I could have saved myself time, money, and frustration if someone had explained to me the issues surrounding this topic?

4. Am I constantly correcting misconceptions regarding this topic?

5. Am I considered to be an expert in this field? For example, do I present my research or experience at conferences?

If your answer is "yes" to any of these questions, your topic is probably of interest to OSBR readers.

When writing your article, keep the following points in mind:

1. Thoroughly examine the topic; don't leave the reader wishing for more.

2. Know your central theme and stick to it.

3. Demonstrate your depth of understanding for the topic, and that you have considered its benefits, possible outcomes, and applicability.

4. Write in third-person formal style.

These guidelines should assist in the process of translating your expertise into a focused article which adds to the knowledgable resources available through the OSBR.

Upcoming Editorial Themes

August 2008	Education
September 2008	Social Innovation
October 2008	Building Community
November 2008	Health and Life Sciences
December 2008	Enabling Innovation

Formatting Guidelines:

All contributions are to be submitted in .txt or .rtf format and match the following length guidelines. Formatting should be limited to bolded and italicized text. Formatting is optional and may be edited to match the rest of the publication. Include your email address and daytime phone number should the editor need to contact you regarding your submission. Indicate if your submission has been previously published elsewhere.

Articles: Do not submit articles shorter than 1500 words or longer than 3000 words. If this is your first article, include a 50-75 word biography introducing yourself. Articles should begin with a thought-provoking quotation that matches the spirit of the article. Research the source of your quotation in order to provide proper attribution.

Interviews: Interviews tend to be between 1-2 pages long or 500-1000 words. Include a 50-75 word biography for both the interviewer and each of the interviewee(s).

Newsbytes: Newsbytes should be short and pithy--providing enough information to gain the reader's interest as well as a reference to additional information such as a press release or website. 100-300 words is usually sufficient.

Events: Events should include the date, location, a short description, and the URL for further information. Due to the monthly publication schedule, events should be sent at least 6-8 weeks in advance.

Questions and Feedback: These can range anywhere between a one sentence question up to a 500 word letter to the editor style of feedback. Include a sentence or two introducing yourself.

Copyright:

You retain copyright to your work and grant the Talent First Network permission to publish your submission under a Creative Commons license. The Talent First Network owns the copyright to the collection of works comprising each edition of the OSBR. All content on the OSBR and Talent First Network websites is under the Creative Commons attribution (http://creativecommons.org/licenses/by/3.0/) license which allows for commercial and non-commercial redistribution as well as modifications of the work as long as the copyright holder is attributed.

The Talent First Network program is funded in part by the Government of Ontario.

The Technology Innovation Management (TIM) program is a master's program for experienced engineers. It is offered by Carleton University's Department of Systems and Computer Engineering. The TIM program offers both a thesis based degree (M.A.Sc.) and a project based degree (M.Eng.). The M.Eng is offered real-time worldwide. To apply, please go to: http://www.carleton.ca/tim/sub/apply.html.