OSBR.CA

The Open Source Business Resource

JULY
2009

JULY 2009

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

The editorial theme for the July issue of the OSBR is "collaboration". While online collaboration has been a hallmark of open source software (OSS) communities, the articles in this issue demonstrate that open collaboration extends far beyond the creation of software. The authors discuss diverse collaboration opportunities including: brainstorming across disciplines, social innovation, aggregating non-profit donations, the green environment movement, open educational resources, introducing students to communities, and managing single-user software applications.

As always, we encourage readers to share articles of interest with their colleagues, and to provide their comments either online or directly to the authors. We hope you enjoy this issue of the OSBR.

The editorial theme for the upcoming August issue of the OSBR is "tech entrepreneurship" and the guest editor will be David Hudson from Lead to Win. Submissions are due by July 20--contact the Editor if you are interested in a submission.

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 anddDNSStuff.com and is the author of the books BSD Hacks and The Best of FreeBSD Basics.

Recently Sir Tim Berners-Lee, inventor of the World Wide Web, was asked whether the future of innovation lies in collaboration. Before giving you his answer, let's consider the question ourselves.

Around the world, people collaborating using Berners-Lee's invention have achieved results too time consuming or expensive to accomplish otherwise. An open source operating system (Linux) and the world's largest encyclopedia (Wikipedia) are but two early, well-known examples.

It may be that the size of the challenges we face gives us no choice but to collaborate, that it is an adaptive response engaging us in the necessary work of reshaping our institutions and our societies in a time of global transition. The call to high purpose is a compelling one, and the evidence is that the response is being heard in many quarters. Collaboration can be fun, and it can also be frustrating.

This issue of the OSBR takes a closer look at these ideas. The papers assembled here, three of which were written collaboratively, look at the nature of collaboration itself. They introduce new ideas and tools for open source collaboration; and examine how collaboration on open source and open content is changing formal education.

Citing inventors from Gutenberg to Darwin, author and teacher Joseph Wilson reminds us that innovative ideas often come from collaboration across disciplines. The Treehouse Group, of which he is a co-founder, conducts highly engaging and enjoyable events to elicit the creative contributions of diverse audiences, with results that are often astonishing.

Anil Patel and I propose 'Applied Collaboration Studios' as a means of transforming the social sector. In our model, open source technologies and social process tools would support new 'collaboration platforms', involving private and public sector partners in generating continuous social innovation.

Peter Deitz and Christine Egger describe how Social Actions came into being, and explore what lies ahead for this database that aggregates information from various on-line portals, making it possible for programmers to create a new generation of tools for social engagement.

Evan Andrews, an analyst with Sylvatica, a Life Cycle Assessment consultancy, introduces the Earthster project and issues a call for open source tools to map the ecology of industrial processes – a fundamental step in our efforts to build a sustainable economy.

Two papers consider the use of open source and open content in formal education systems. In the first, Norm Friesen outlines the history of such efforts, and contrasts the opportunities and challenges inherent in creating learning resources using wikis, with an open courseware approach that emphasizes wide sharing of course materials and technologies.

In the second paper, Chris Tyler describes how he teaches students to participate in solving real world problems by having them contribute to open source communities around the Mozilla project and OpenOffice.org.

Finally, Andy Adler, John C. Nash and Sylvie Noël introduce TellTable, an open source framework they have developed that allows single user applications to be used collaboratively and with ease.

Together, the papers in this issue provide an answer to the question posed to Sir Tim Berners-Lee – one with which he concurred. It is a decided 'yes'!

Stephen Huddart

Guest Editor

Stephen Huddart is the Vice President of The J. W. McConnell Family Foundation in Montreal and the Director of SiG (Social Innovation Generation) @ McConnell.

"…it is the interaction between data that causes change. The fundamental mechanism of innovation is the way things come together and connect."

James Burke, The Pinball Effect

Anecdotal evidence suggests that truly innovative ideas and successful adaptation to market conditions comes from collaboration with people across traditionally demarcated fields of study. In science, economics, and business, it is new ideas that are imported from other realms that are most successful in affecting change.

The Treehouse Group (http://treehousegroup.org) is a Toronto-based collective of thinkers from a wide variety of backgrounds that seek to subvert the traditional notion of what constitutes a conference or a networking opportunity by using a wide variety of brainstorming techniques and cross-disciplinary activities to engage participants.

Ideas at the Intersection

In the early 18th century, a wave of immigrants hit Britain's coast. French Protestants, known as Huguenots, settled in an area just outside the old City of London known as Spitalfields. The market at Spitalfields and the nearby Petticoat Lane initially focused on the Huguenot specialty of weaving, but soon grew into a hub of intellectual exchange of all sorts. One young weaver, inspired by the advances made in Enlightenment science, turned his skill at manufacturing silk towards the fledgling field of lens-making.

John Dollond was inspired by the way the lenses in our eyes focus light with such precision. He combined concave and convex lenses in a way that resulted in near-perfect images, eliminating "chromatic aberration" (http://en.wikipedia.org/wiki/Chromatic_aberration). In 1781, this technology, a boon for manufacturers of eyeglasses was in turn scooped up by London astronomer William Herschel, a regular at the market, to create a telescope powerful enough for the discovery of the planet Uranus.

James Burke (http://en.wikipedia.org/wiki/James_Burke_(science_historian)) has documented such connections between people and ideas for years through books such as The Pinball Effect, his BBC television show "Connections" and his column in Scientific American. He has recently launched an on-line educational initiative called the Knowledge Web (http://k-web.org) where students can surf their way through a web of interconnected innovations in science and technology. Burke is a master at uncovering the interconnected web of ideas and technologies that, when allowed to work together and ferment, result in real and lasting change in our world.

Burke is, however, a historian. Tracing the web of social change is easy in hindsight, but near impossible in the present. How do we evaluate the importance of a new technology or a best-selling book? The answer lies not in the intrinsic value of the book's thesis, or how many people buy a widget, but in how it connects to other spheres of influence.

The historical record shows us that truly innovative ideas do not arise in isolation from one another, but at a place like Spitalfields market, a place where one field of study like silk weaving turns into something else, like astronomy. A good idea is not a static, containable thing, but a connector: a burr that latches onto other people and their projects, changing things in the process.

Frans Johansson explores this idea in his book The Medici Effect: What Elephants and Epidemics Can Teach Us About Innovation (http://themedicieffect.com). He points to the patronage of the

influential Medici family in Renaissance Florence as a force for unbridled creativity. The family funded intellectual exploration wherever it lead, in a spirit that encouraged the crossing of fields of study. Leonardo DaVinci is often held up as the model of this era, as an accomplished engineer, artist, anatomist and musician, a thinker that worked at "the intersection." The intersection is a place where disciplines meet, where boundaries between fields of study collapse, revealing a new intellectual landscape.

"One thing we know about creativity," says Marc Tucker, Head of the Washington-based National Center on Education and the Economy (http://www.ncee.org), "is that it typically occurs when people who have mastered two or more quite different fields use the framework in one to think afresh in the other." Think of the now famous theory that the impact of an asteroid killed off the dinosaurs. It was not proposed by a palaeontologist, but by nuclear physicist Luis Alvarez who had an interest in astronomy.

Charles Darwin, for all his momentous effect on the world of biology, was not a trained biologist. His background in geology allowed him to think deeply about how things change over time. His intellectual curiosity brought him out of his field of study and onto the deck of a ship that travelled the world in search of the new. Upon his return, it was his collaboration with zoologist John Gould that allowed him to propose his revolutionary theory of natural selection.

The Fallacy of Group-Think

We need people outside our fields to collaborate with. Otherwise, companies and social organizations risk falling into the trap of "group-think," where bad ideas are reinforced from within through an iterative process of self-reinforcement. Most famously, NASA found group-think to be one of the factors of the Columbia disaster of 2003. Insulation tiles on the wing were damaged by falling ice during lift-off, yet the Mission Management Team (MMT) discounted reports of critical damage and came to the conclusion that even if there was damage, "nothing could be done."

In his book The Wisdom of Crowds (http://en.wikipedia.org/wiki/The_Wisdom_of_Crowds), James Surowiecki details the process by which the Columbia Accident Investigation Board (CAIB) uncovered the MMT's decision-making process. What the group lacked was the "cognitive diversity" to encourage disparate opinions that might have brought the astronauts home safely. The elimination of new perspectives was a result of the group's adherence to the strict hierarchical structure that had become entrenched at NASA.

What the team needed was a reminder that solutions to tough problems don't often occur within the confines of fields of study, but at the margins. The easiest way to access these margins is to open up the process to collaboration and discussion with as wide a range of people as possible. In other words, open up your decision making to the wisdom of the crowds. Collectively, the vast and varied experiences of a large group can provide much-needed advice on how to proceed in sticky situations, much more than the experience of any one individual.

In our current economic quagmire, it has become a truism to appeal to innovation and "outside-the-box" thinking to allow companies to survive. But organizations that are not practised at this will struggle. They will hire the same consultants and read the same industry analyses and demographic studies without ever bumping up against the sides of their boxes, let alone break through.

The failure of General Motors is a classic example of a company that got too big and became too entrenched in their own way of thinking to contemplate change. Instead of struggling to save their company through growth and power over government agencies with their hands on bail-out funding, CEOs might have benefited from a 180 degree turn. Leafing through the ancient Taoist text, the Tao Te Ching, we find stanzas 182 and 183:

Grass and trees are pliant and fragile when living,

But dried and shrivelled when dead.

Thus the hard and the strong are the comrades of death;

The supple and the weak are the comrades of life

The strong and big takes the lower position,

The supple and weak takes the higher position.

The Taoist philosophy of flexibility in the face of adversity is the same idea many economists are now espousing to survive the recession: split your company up into smaller chunks so they can adapt more readily to market forces.

It's a safe bet that many CEOs took a comparative religion course as part of their humanities undergraduate degree. But books not directly related to the handling of multi-national companies were seen as a waste of time, a frivolous diversion.

In university, intellectual playfulness is more accepted. Pursuing esoteric lines of thought is expected and embraced, especially in the humanities. "Knowledge for knowledge's sake" is a common mantra for defending the public good of universities. After graduation, it sometimes feels like we've entered a period of intellectual stagnation, surrounded by people with the same skill sets and experiences.

The collaboration between Darwin, Gould and many other scientists and philosophers during the 19th century was called "philosophical laughing," by Charles Darwin's grandfather, Erasmus Darwin, himself a fan of intellectual banter. Darwin and his colleagues were able to pursue knowledge just for the fun of it, smiling at each other's preposterous ideas along the way.

Science shows us that the human brain is evolved to be remarkably adaptable to new ideas and conditions, but only if we embrace new experiences. In The Brain That Changes Itself, Norman Doidge surveys the burgeoning field of "neuroplasticity," a field devoted to studying how the brain is able to rewire itself in new situations (http://www.normandoidge.com).

When children are born, they enter the world with an "undifferentiated" brain, a seething mass of firing neurons that eventually get pruned and trimmed into a map that corresponds to the world in which they live. This period of time is crucial for brain development. Children who miss out on key periods of social, linguistic or emotional development retain cognitive behaviours that become locked in as they grow into adulthood. Children are also creative, as their brains work overtime, experimenting with new connections and neural networks.

"All people start out with plastic potential," says Doidge. "Some of us develop into increasingly flexible children and stay that way throughout our adult lives. For others of us, the spontaneity, creativity, and unpredictability of childhood gives way to a routinized existence that repeats the same behaviour and turns us into rigid caricatures of ourselves."

Recent studies show that we can regain this plasticity of our youth. As our brains get used to firing the same neuronal connections day after day, they become more resistant to change. Yet brain scans of adults who make an effort to engage in new experiences show evidence of massively reworked brain maps. Adults who learn a new language, take a drawing course, or otherwise challenge themselves intellectually can more easily adapt to the rapidly changing world around them, and even have lower risks of dementia and other health problems. Plasticity begets plasticity. When we have a new experience, or learn something new, the human mind shuffles the data around and works it into previously learned experience. New knowledge doesn't grow dust and remain static, but gets parcelled up and redistributed, used as the building blocks of new ideas. If we close ourselves off from the new, we risk stagnation in our jobs and lives.

In his book Proust was a Neuroscientist, Jonah Lehrer (http://jonahlehrer.com) recounts stories from the 19th century of artists uncovering truths years before scientists. Working in the same cities and mixing with the same people allowed artists and scientists to uncover parallel truths. Unfortunately, many scientists in the 19th century, enamoured with the power of positivism, scoffed at the subjective experiences of the arts.

Auguste Escoffier (http://en.wikipedia.org/wiki/Auguste_Escoffier) was a famous Parisian saucier who invented veal stock at the turn of the 19th century. Escoffier emphasized the importance of stock for revealing tastes within meals at the same time as a Japanese biochemist isolated the amino acids that made meat taste so good: he named this taste umami, the Japanese word for delicious. Scientists in Europe were more sceptical. They refused to believe in Escoffier's new mode of cooking, because they were convinced people could only perceive four tastes: sweet, salty, sour and bitter.

The power of group-think was as strong in 19th century biochemistry as it was in 20th century NASA or 21st century General Motors. Imagine how things might have changed if the biochemists had invited a cook to their conference instead of another scientist.

Enter the Treehouse

Founded in 2006, the Treehouse Group (http://treehousegroup.org/) is a collective of Torontonians devoted to embracing this idea of cross-disciplinary collaboration. Inspired by the prospects of living in a diverse and dynamic city, the Treehouse Group organizes conferences, monthly brunches, science-fairs, and educational sessions dedicated to exploring that fuzzy and exciting region where fields of study overlap. This is where truly creative ideas foment.

Our flagship series, the Toronto ideaXchange, has brought together hundreds of people from different fields of study to grapple with social problems and play with ideas. We've witnessed conversations between lawyers and musicians, home contractors and information technology professionals, entomologists and high-school students.

We see narrowing of perspective in our jobs every day. In my experience, sitting in a lecture hall at a conference full of people that share your specialty results in one of three outcomes: i) boredom; ii) a feeling of defensiveness if you disagree with the presenter; and, iii) if you're lucky, a notebook full of good ideas you'll never open again.

Millions of dollars are spent every year to provide employees with opportunities for professional development. How can we tweak the structure to infuse events with

the creativity and ingenuity we need to solve our current problems? How does the Treehouse Group respond?

1. Against boredom: at almost every talk I've been to, a speaker is announced, the PowerPoint is fired up and the speaker is quickly talking to the eyelids of the audience. At many Treehouse events, PowerPoint is all but banned. If people are presented with something they aren't expecting, their attention is immediately captured.

Or if PowerPoint does sneak its way in, it's in the form of "PowerPoint karaoke", where small groups are given the same set of graphs, slides and photographs, and have 20 minutes to rearrange them into a presentation for the rest of the participants. People are never bored during this activity.

Imagine you settle into a lecture hall and are presented with a graffiti artist creating a large mural before your eyes, or a musician demonstrating an organ that works not through bellows and tubes, but through flowing columns of water. At our first event in 2007, I witnessed John Evans, Chair of the Canada Foundation for Innovation (http://innovation.ca), concentrate on performing a drum-roll, a skill he was exposed to only a few minutes earlier at the MaRS Discovery District (http://www.marsdd.com). At an event this January, Deb Matthews, Provincial Minister of Children and Youth Services, was taught how to scratch a record by a professional disk jockey.

We call these activities brain cleansers. They are a way to grab the audience's attention and clear their mind much like water cleanses the palette of a sommelier. After these activities, collaboration has renewed vigour and ideas flow more freely. Another way of subverting people's expectations is with location. Try having your board meeting at a park, or an elementary school, or a fire station. Last June, the Treehouse Group had a meeting in the middle of Bloor Street during a street festival that closed the street down (http://bigonbloor.com). We invited members to bring food for a giant potluck dinner where we mashed up some ideas on a white board.

Earlier in the spring, we hosted our annual Grown-Up Science Fair, where participants made flubber, played periodic table twister, played with science overheads from the 1960s and debated the merits of the new video game Spore. No notes were taken, but there was more intellectual energy in that room than at the last 10 conferences I've attended.

2. Against defensiveness: it is easy to tell people to "have an open mind" and "think outside the box," but people need prompts. When we hear people talk about subjects we have been studying, a common response is to mentally joust with the speaker and reject and argue away the points they are making. This is a natural way for people to protect their reputations and their egos. The trick is to present people with something they don't have a well-entrenched opinion about.

At a recent event, the Treehouse Group was given the task of providing an evening of orientation for 30 energetic teenagers in Toronto. The students, here from all over the world, were set to embark on a two week tour of the Arctic through an organization called Cape Farewell (http://www.capefarewell.com), to explore issues of climate change through science and the arts. The students were ambassadors of environmentally responsible living and exuded confidence and knowledge.

To turn the tables, we asked them to role-play as one of a number of professionals such as an oil executive, a politican from China, or a First Nations activist. We gave

them 60 minutes to come up with a climate change treaty that they could all sign in good conscience. The activity was not easy. There were people yelling and getting frustrated, there were groups who refused to sign. There was even one corn farmer who got his "union" to "strike" until the government refused to do any more business with China.

But there were also some genuinely creative solutions. One group focused on getting their message out through the media, while another decided to hold a referendum so the citizens of their countries could rank the priorities of dealing with climate change. When the students were asked to role-play, their empathy shot up and their defensiveness shot down, crucial for consensus-based problem solving.

3. Against taking notes: studies in education show that around 20% of the population are auditory learners. That means that most of us need to get up and interact with a subject in order to understand it. Taking notes at a conference helps retention but is a poor way to internalize new concepts. Activity and engagement are what stick.

At our monthly brunch meetings, one of our favourite activities is the "Great Magazine Mash-up." Participants grab a magazine they don't normally read from a pile. They open it at random and try to combine whatever is on the page in front of them with the subject of their neighbour's magazine.

At an ideaXchange recently, the Director of the Pathways to Education organization in Regent Park found himself staring at an article about owls. Next to him, a high school student from Étienne Brûlé Secondary School in Toronto found an ad for an interior design company. The group came up with a business proposal for an architecture firm specializing in animal treatment facilities, a company that designs houses for people who rehabilitate birds, and an interior design company that specializes in natural colour palettes based on the colouring of owls.

Not all of the ideas are winners, but enough completely new concepts are created in a short period of time where forward thinking people can go back and revisit conversations that have real potential to affect change. Leaving time for drinking wine and swapping business cards after the hard work is over is crucial to the success of Treehouse events.

Final Thoughts

If you find yourself at your next board meeting drifting off to sleep, ask yourself if you can think of a truly engaging experience that would expand people's horizons and inject some sorely needed ingenuity into the standard model of business. To avoid falling into the trap of group-think, we need to embrace intersections with other disciplines, and have some fun. In a 2002 article in the New Yorker, Malcom Gladwell said it best: "losing sight of what you truly believed when the meeting began is one way of defining innovation."

Joseph Wilson (http://josephwilson.ca) is co-founder of the Treehouse Group and a teacher at a Toronto high school. He is also a freelance writer, focusing on issues in science, technology and culture. His work has appeared in NOW Magazine, the Toronto Star, Spacing Magazine, and The Globe and Mail, and he is working on a Grade 9 Science textbook for Nelson Education. He graduated from the University of Toronto with an Honours Bachelor of Science in Astronomy and Semiotic & Linguistic Anthropology, and a Bachelor of Education from the Ontario Institute for Studies in Education, specializing in social justice and global education.

"Our image of the world now, constructed by people we once thought we could rely upon for such work [...] is actually and philosophically false. It's time to replace it with an image that actually works. What we need is a framework for the sort of change that fits our world – and that lays a foundation for the widespread personal involvement of millions of people that will make such change useful, durable and sustainable."

Joshua Cooper Ramo
http://joshuaramo.com

This paper asserts that the voluntary or social sector plays a pivotal role in generating and disseminating social innovations through collaboration with diverse partners. The authors explore the potential to engender a quantum leap in the sector's efficiency, reach, and impact through the combined use of open source technologies, social process tools and collaboration platforms. The objective is to contribute to a new generation of intelligent social systems, enabling an evolutionary recalibration of relationships among ourselves, our social and economic institutions, and the planet.

As a means of integrating and disseminating the most promising approaches, the concept of 'Applied Collaboration Studios' is proposed. Their primary activities would be: i) dissemination of open source technologies; ii) ongoing instruction and coaching in the use of social process tools; iii) the assembly and launch of multiple collaboration platforms; and iv) collaboration with other like initiatives to create ecologies of scale that inform and precipitate systems' change toward greater resilience.

The paper concludes with a reflection on the conditions necessary for such a project to come into being as an open source initiative, and an invitation to contribute to an ongoing discussion at http://www.appliedcollaboration.org.

Why We Need Applied Collaboration Studios

In Platforms for Collaboration (http://nambisan.typepad.com/satish_nambisan/2009/05/platforms-for-collaborative-social-innovation.html), Satish Nambisan argues that: "Organizations must look beyond their own boundaries. Adopting [a] network perspective forces them not only to consider how their agendas fit with broader social problems, but also to develop the skills for collaborating with diverse partners."

Our continuing efforts to adjust human economic and social behaviour so that is equitable, enjoyable and environmentally sustainable begins with a dual premise:

1. We have to learn quickly to do a lot more with a lot less.

2. Getting to dynamic equilibrium requires high levels of ingenuity and collaboration both within and across traditional sectors and silos.

We focus on what this implies for the social sector in Canada.

Getting Past Social Sector 1.0

The initial goal is to overcome bottlenecks in communications among donors and grantees, focused on planning, administration and reporting within the sector. This is equivalent to looking at an operating system and not the programs that run on it. In other words, how the sector functions, not what it does. If funders and grantees used the same quantitative metrics and common matrices for categorizing and sharing qualitative data, in order to 'speak the same language', the result would be less waste, and increased capacity to collaborate.

The recent paper Drowning in Paperwork: Distracted from Purpose (http://projectstreamline.org/downloads.php) documents how organizations misspend considerable human and financial resources writing numerous grant applications and reports to funders. One answer is to introduce standard reporting formats and data repositories, as the Cultural Data Project (http://www.culturaldata.org) is doing. When grantees input their operational information once, so it can be used in application and reporting processes to multiple funders, they save time and money. Tailored reports can be generated by individual funders, using subsets of the available information. Another paper, The Non-Profit Marketplace: Bridging the Information Gap (http://www.givingmarketplaces.org/materials/whitepaper.pdf), redefines the information needed to assess operational efficiency and social impact. By analyzing such data, up-to-date comparisons can be made, and resources directed to organizations that perform best. What is currently a plethora of reporting protocols could be replaced by a few, adapted to specific fields and types of activity.

Designing, testing and implementing such a system, with input from multiple stakeholders, would make an exemplary project for Applied Collaboration Studios.

Over time, Studios would build a growing, living repository of open source technologies applicable to social sector organizations, annotated by user groups using wikis, and documented in case studies and outcome reports. They could organize workshops and coaching in the use of specific tools, in connection with projects. If warranted, the Studios might collaborate in funding new work.

For example, Social Actions (http://socialactions.com), which aggregates information from diverse online sources into an open application programming interface (API) database, could collect and present information on volunteering opportunities across Canada, enabling these to be presented in a manner similar to what Canada Helps (http://canadahelps.org) does with online donations.

Adding Information Dashboards with Open APIs

In order to make smarter decisions, better information is required about the state of particular issues or domains, along with the ability to observe trends and changes. Optimal operation of Applied Collaboration Studios requires that current data be organized into relevant sets or 'dashboards'. Indicators that can be assembled, displayed, and updated allow participants to work with common reference points, to review the known and unknown, and to make decisions based on shared information.

Tools like Vital Signs (http://www.vitalsignscanada.ca) are demonstrating high utility in framing civic discussions, guiding philanthropic contributions and contributing to public policy discussions. It would be interesting to see it applied by universities or schools at a neighbourhood level. Another promising collaboration might occur with the Community Leadership (http://www.cclnet.org) programs that exist in many Canadian cities.

The Canadian Index of Wellbeing (http://www.ciw.ca) is a set of indices whose objective is to improve on the Gross Domestic Product as the measure of social, environmental and economic health. Three of eight sub-components are ready now, offering comprehensive datasets with open APIs, enabling use by the social sector.

Introducing the Next Generation of Social Process Tools

Social process tools mimic open source technologies in that they awaken collective intelligence, enable mass collaboration, cost little and frequently produce extraordinary results. Examples of such tools include World Café (http://www.theworldcafe.com/), Appreciative Inquiry (http://appreciativeinquiry.case.edu/), Deliberative Dialogues (http://www.cprn.com/doc.cfm?doc=1238&l=en), and Future Search (http://futuresearch.net). Such initiatives enable groups of diverse individuals with varied capacities and interests to develop shared understanding, and draw on wellsprings of compassion and creativity to forge new ideas and directions.

Chevalier and Buckles have published SAS2: A Guide to Collaborative Inquiry and Social Engagement (http://www.sas2.net/en/sas2-guide/). SAS stands for Social Analysis Systems and the SAS2 website outlines some 20 such processes. The tools are available online and the authors are applying them to a range of projects, while training a cohort of practioners in their use.

Recently, SiG @ Waterloo (http://www.sig.uwaterloo.ca) tested several new process tools with social sector organizations involved in disseminating social innovations. A brief review is sufficient to hint at the potential such tools have for improving collaborative work on complex issues.

1. As a means of providing context for thinking about dynamics in systems and formulating strategies appropriate to four distinct phases in a transformation process, Frances Westley presented an updated version of the Panarchy framework (http://www.resalliance.org/593.php).
2. Brenda Zimmerman introduced a 'multiple lenses' tool for examining issues from diverse perspectives. This provided several participants with fresh insights into a particular challenge, changed conversations among stakeholders, and led to several new strategies for collaboration.
3. Angela Dumas led a Totemics session (http://tinyurl.com/mqpwev) to define an organization's essence through the guided co-creation of visual metaphors. In one case, this helped diverse stakeholders interested in working together to achieve passionate consensus.

The initial work of developing such tools is resource intensive, involving collaboration among academics, process consultants, philanthropists, and others. Collaboration Studios propose to serve as clearing houses for such tools and as a working laboratory for their application and refinement.

Applied Collaboration Studios

John Chambers, CEO of Cisco Systems, once stated: "Without exception, all of my biggest mistakes occurred because I moved too slowly. The future is about collaboration and teamwork and making decisions with replicable process that offers scale, speed, and flexibility".

Having proposed technological and social process tool innovation, the requisite capacities are in place for collaborations at scale involving a diverse range of actors, including social sector leaders, thought leaders, policy makers, academics, artists, students, and business leaders. For the sector to engage in the larger project of social transformation, efforts must converge around collaboration. We are attracted to the notion of Studios as a space for collaboration and learning.

Multiple projects can take place there, in a variety of media. Mentors would be available, and it is possible to learn just by observing, or to become involved in co-creation. Work would be produced for various publics, and Studios recognized as a source of ideas and interventions that are both practical and elegant.

Satish Nambisan suggests three types of collaboration platforms:

1. Exploration: what is the problem?

2. Experimentation: what is the solution?

3. Execution: give the solution away.

Taking these as templates, we can cite three practical examples within one metropolitan area to illustrate the Studios' role in changing the way social issues are addressed. These examples are explained below then summarized in Figure 1.

Youth Challenge Fund (http://www.youthchallengefund.org) recently granted $40 million dollars to 200 youth-led initiatives in Toronto's underserved communities. A majority of these initiatives are small, without much capacity or expertise to manage their growing operational infrastructure. Studios could bring groups together to **explore** the common challenges these organizations are facing and, based on needs analysis, assemble a suite of the most appropriate tools for planning, administering and reporting on their projects, and adding coaching and evaluation.

Studios would support **experimental** forms of collaboration. Business for the Arts (http://www.businessforthearts.org) is proposing a shared event calendar for art gala events run by their network of partner charities.

Events such as the Royal Ontario Museum's Prom, the Art Gallery of Ontario's Massive Party, and the Science Center Innovator's Ball all target a similar audience: young professionals who will be the next generation of board members and donors. Event planning, marketing, awareness and other logistics between initiatives are currently uncoordinated, siloed, and occasionally in conflict. Studios could bring the organizers together to model user needs and implement solutions. A shared planning and event calendar is one simple first step, with various forms of collaboration around purchasing and analysis a subject for further work.

A final example is a collaboration designed for **execution**. Eva's Phoenix Printshop (http://www.evasinitiatives.com/ephoenix.php) is a social enterprise that has provided pre-apprenticeship and employment training for over 175 at-risk youth. Eva's in-house commercial printshop provides both a revenue stream to support the program, and on the job experience for its clients. A recent poll of 50 Toronto-based charities acknowledged they had heard of Eva's Printshop, but only one had used its services. Studios' project planning capacity could uncouple the pre-production bottleneck between charities working on shoe-string design and assembly budgets and Eva's quoting and print-run capacity. They could then design a system for routing a percentage of stakeholders' print work through Eva's. The win-win is apparent: charities can claim deeper blended-value in their work while Eva's builds a sustainable client base.

Figure 1: Three Types of Collaboration Platforms

	Exploration	Experimentation	Execution
Example	**Youth Challenge Fund** *"Lots of money granted to small, grass-roots organizations"*	**Business for the Arts** *"Coalition is developing a shared calendar for gala art events"*	**Phoenix Print Shop** *"Opportunity to connect charities print needs to social enterprise print shop"*
Objective	-Define the core problem -Connect with problem solvers	-Develop solution prototypes -Test prototypes in near-world contexts	-Build and disseminate solution templates -Help adopters adapt to system-wide changes
Role of Studio	-Build a diverse coalition of stakeholders -Give stakeholders numerous and varied forums to air their concerns -Identify potential problem solvers	-Integrate ideas from diverse stakeholders -Offer neutral environments for deep testing of solutions	-Facilitate the collaborative development and diffusion of solution templates -Provide resources that adopters can use to manage the "rippled effects" that follow implementation
Desired Outcomes	-Shared definition of problem -List of potential solutions	-Assessments of possible solutions -Solutions recommendations	-Solution templates & Implementation of standards -Rapid adoption of the social innovation

Adapted from *Platforms for Collaboration*, Satish Nambisan. Stanford Social Innovation Review Summer 2009

Collaboration at Scale: The Framework Foundation's Expansion across Canada

Framework Foundation (http://frameworkfoundation.ca/) is the organizer of Timeraiser (http://timeraiser.ca). This organization increases civic engagement among young professionals via an innovative arrangement among companies, emerging artists, and social sector organizations.

Since 2004, this program has generated 45,000 volunteer hours, invested $260,000 in the careers of emerging artists, engaged 3,600 people to volunteer for causes and worked with 250+ charities in six Canadian cities. Timeraiser is now well positioned to reach many more cities effectively. However, this was not always the case.

Several years ago, Framework Foundation was approaching the scalability wall. Staff and volunteers struggled to keep pace with information requests from various stakeholders such as donors wanting pledge completion data, agencies asking for demographic analysis, the board requesting budgets, artists wanting to know the location of their art, volunteers wanting information about engagement opportunities, and the media wanting examples of successful matches.

As the volume of requests grew, so did the pressure to answer each in a timely fashion. Since key contacts changed frequently, a whole re-education process was required. These pressures were exacerbated by the lack of time and resources to implement an integrated information management technology platform.

At a retreat, staff and volunteers whiteboarded the complexity of the Timeraiser stakeholder mix. Then they considered what types of information would be needed to satisfy a majority of requests, including what was unique to each stakeholder and what was generic. This exercise also factored in the 12 unique workflows and 100 document templates required per Timeraiser event.

At the conclusion of this exercise, Framework Foundation had determined where the data and communication bottlenecks would occur as the organization grew. To manage the growing complexity of its operations, Framework had to select technological tools that enabled agile project coordination and robust relationship management. Then, instead of constantly evolving into new states of instability as often occurs when organizations scale up an initiative, Framework would enjoy increased capacity to manage new projects it was interested in developing.

After a review of off-the-shelf and cloud computing options, Framework decided on the integrated features of Google Apps (https://www.google.com/a/) and http://Salesforce.com, which is free for charities. The customized email domain names, document management, calendar synchronization and wiki-style portal sites provided by Google Apps effectively complemented Salesforce.com's relationship management functions. As well as improving the volunteer experience that produces Timeraisers in a growing number of cities, the result is that new projects can now be launched in days, instead of weeks or months.

Framework Foundation was able to design and launch one new idea in hours. Upon reading Colleen Kelly's A People Lens (http://www.volunteerinc.com/pdf/A_People_Lens.pdf), a hypothesis emerged: "while most leaders in Canada's nonprofit and voluntary sector community agree that people - staff and volunteers - are an organization's most important asset, most post-secondary curriculum focuses on fundraising." As a result, staff and volunteers quickly created a Google site (http://courseresearch.timeraiser.ca), listed the 25+ post-secondary institutions for curriculum review, constructed a methodology, enlisted volunteers to review course outlines, completed the analysis and immediately published the results for the world to see and comment on.

Activities related to this project were tracked and monitored in Salesforce.com. Volunteer hours were recorded, the Google docs dynamically linked as attachments and email communication logged in a coordinated fashion. Staff or board members with the correct permission levels can instantly view the status of any project and the contribution from anyone outside the organization .

Had Studios been in place during Framework Foundation's exploration and experimentation stages, they would have provided guidance through the following steps:

- identify the stakeholder mix for a particular organization, issue or domain
- within that mix, envisage how information needs to flow
- expand the visioning exercise to determine where potential bottlenecks may exist
- based on where bottlenecks may appear, explore how to best close the communication/information gap with appropriate tools and work-flow design
- test and adapt tools and workflow design in near-world situations

A significant and unintended result is how the project to organize Timeraisers has influenced the staff and board's capacity to see and work beyond Framework's previous borders. Framework now operates consciously in a transition zone between 20th and 21st century working methods. It has discovered that collaboration at scale, using what Clay Shirky calls the 'power of organizing without organizations' (http://herecomeseverybody.org) is optimal for addressing complex issues. It releases financial and human capital for new purposes, and pays an unexpected dividend in affording perspectives on the work in a larger context.

Collaboration Among Collaborators

Timeraisers involve a novel linking of institutions to address the issue of declining volunteerism among young professionals.

Similarly, Applied Collaboration Studios would serve as a collaboration platform for designing innovations around other complex challenges. In some cases, these might be time-limited projects with specific, planned outcomes. More often, they would be structured as systems of continuous innovation, enabling the ongoing testing and refinement of tools and strategies, continuous updating of performance indicators, and investment in successive generations of new ideas and approaches.

Similar efforts are underway elsewhere, and offer possible links to those emerging in Canada. In fact, since social innovation is an emerging global phenomenon, it makes sense to prioritize Canada's potential contributions to the field. Here are some examples of large scale collaboration platforms:

Ashoka Changemakers (http://ashoka.org/changemakers), a process for open sourcing social innovation, turns the generation and identification of novel solutions into competitions in which ideas are typed according to strategy and sorted according to which aspects of a complex problem they address. Using this common framework, all options become visible and up for discussion, making it possible for anyone to suggest improvements or link ideas to create hybrids. By keeping the prizes small (typically $5,000) and the process fun, emphasis is placed on supporting a healthy exchange of ideas. Being an open process reduces the reinventing the wheel phenomenon, another drag on innovation. It provides the philanthropic marketplace with greater fluency around interventions in large systems. In the past year alone, over $30,000,000 in new funding has been directed to projects identified in the competitions.

Another model to consider is 'The Lab' (http://www.nestalab.org.uk/), created by the UK National Endowment for Science, Tecnology and the Arts (NESTA), whose mission is to introduce public sector innovation. The Lab is organized into sections called Challenges, Methods and Learning, and incorporates input from governments, and the private and social sectors.

Social Innovation Exchange (SIX, http://socialinnovationexchange.org) is a global learning and exchange program for social innovators.

Social Innovation Generation (SiG, http://sigeneration.ca/), a partnership among MaRS Discovery District, the University of Waterloo, the PLAN Institute, and The J. W. McConnell Family Foundation, is positioned to play a role in advancing this work in Canada, along with other allies.

Conclusions

While market failures in the private sector currently take up the lion's share of media attention and attract massive financial support from government, it is within and around the social sector that many of the solutions to our present crises will be found. Using open source technologies and social processes, Applied Collaboration Studios would host collaboration platforms to improve social sector performance, and structure and implement systems of continuous social innovation.

Projects within the Studios would use open source methods to frame and address a range of challenges, at different levels of scale, making use of dashboards to aggregate and update relevant data. In addition to convening projects around specific challenges, a learning platform would distill and share lessons from across multiple platforms.

Applied Collaboration Studios would seek working relationships with like initiatives globally.

Startup funding would enable the requisite diverse design expertise and mentorship team to be assembled, and a portfolio of demonstration projects to be hosted, some of which already exist in various stages of development.

A scan of where cloud computing and the semantic Web are heading would be helpful, as would a taxonomy and evaluation of social process tools. There are labour and training issues involved too: apart from Web of Change and Social Tech Week, which cover technology and social change, there are no programs that cover these ideas in a comprehensive manner.

We invite readers to visit http://www.appliedcollaboration.org to contribute their ideas and comments.

Stephen Huddart is the Vice President of The J. W. McConnell Family Foundation in Montreal and the Director of SiG (Social Innovation Generation) @ McConnell.

Anil Patel is the founder and Executive Director of the Framework Foundation in Toronto, originator of Timeraisers and the Civic Footprint.

Recommend Resources

Open Sourcing Social Solutions
http://www.mitpressjournals.org/doi/pdf/10.1162/itgg.2007.2.3.125

Volunteer Gateway
http://voluntarygateway.ca

20 Questions Directors of Not-for-profit Organizations Should Ask About Strategy and Planning
http://www.snwebcastcenter.com/data/224/support_doc/FINAL%20English%20StreamingNetwork%20nov%2028.pdf

"It is the nature of thought to find its way into action."

Christian Nevell Bovee

Social Actions (http://SocialActions.com) makes it easier for people to turn their good intentions into meaningful action. The organization has created an open source database of actions people can take on any issue. The actions in the database come from across the social web and include everything from volunteer opportunities to micro credit loans. It currently aggregates opportunities to make a difference from 50+ action sources, including: Canada Helps (http://canadahelps.org), Kiva (http://www.Kiva.org), Idealist (http://www.Idealist.org), Global Giving (http://www.GlobalGiving.org), Give India (http://www.GiveIndia.org), and Greater Good South Africa (http://www.myggsa.co.za). Using the Social Actions application programming interface (API, http://www.socialactions.com/developers/api), we encourage third party developers to build web and mobile applications that intelligently distribute actions from our database on the websites, social networks, and mobile phones that millions of people use every day.

This article describes how Social Actions applies open source principles to the organization's products and processes. In its entirety, Social Actions is intentionally designed to contribute to the ongoing and vibrant conversations about open source practices and principles.

Social Actions' Story

Peter Deitz, Social Actions' founder, started blogging about the organizations involved in online philanthropy in 2006. He quickly recognized that there was no unifying resource for learning more about these websites and the opportunities they offered.

Through the spring and summer of 2007, he wrote to the organizations and requested an RSS (http://en.wikipedia.org/wiki/Rss) feed of the donation opportunities listed on their websites. By the end of August, having gathered close to a dozen of these data feeds, he launched the first prototype of a system that aggregated microphilanthropic opportunities from different sources into a single dataset. The initial website lived in the Drupal (http://www.drupal.org) open source content management system.

During that time, Peter was also blogging about Social Actions and inviting others to join in. By February 2008 there was enough broad interest to create a Social Actions Google group (http://groups.google.com/group/social-actions), where a community of about 20 people generated the beginnings of Social Actions, the organization. Christine Egger joined the project at that time. Her background in the complexity sciences and hermeneutics--both of which emphasize the value of decentralized, context-sensitive, peer-to-peer engagement--are heavily reflected in Social Actions' mission and practice. The Google group attracted a number of mentors and supporters in the nonprofit technology and philanthropy sectors. One such member, web developer Cameron Boothe, volunteered to create a more robust version of the Social Actions prototype. Another, Frerieke van Bree, encouraged Peter to enter Social Actions into three competitions. In May 2008, Social Actions came in third place at the NetSquared Mashup Challenge (http://www.netsquared.org/2008/conference) and the DonateNow Challenge (http://www.netsquared.org/challenges/n2y3/donatenowchallenge), and was selected as a finalist in the Stockholm Challenge (http://www.stockholmchallenge.se/data/2031 2008). These achievements brought critical funds and visibility that fueled Social Actions' development through the summer of 2008.

Over the next several months, a core team composed of Peter Deitz, Christine Egger, Joe Solomon, Jason Mott, Josh Crawford, and Eric Cooper focused on:

- adding more action sources and functionality to the Social Actions API
- developing applications that would serve as compelling examples of how the Social Actions API could be used to distribute opportunities to make a difference across the web
- co-creating a robust open standard for publishing opportunities to make a difference

The process of contributing actions to the Social Actions API is deliberately inclusive and simple (http://socialactions.com/partners/action-sources). Organizations email the RSS feed URL for their actionable content, sorted by date and containing their latest campaigns. Social Actions subscribes to the feeds and includes that organization's profile in its online guide (http://socialactions.com/meet-the-platforms). In lieu of a contract or memorandum, platforms are asked to optionally endorse an online statement as an indication to the public that the organization shares Social Actions' commitment to openness, collaboration, and data portability (http://socialactions.com/platform-endorsement).

In February 2009, Social Actions launched the Change the Web Challenge (http://www.socialactions.com/changetheweb), an online competition to encourage developers to build open source applications that draw on the Social Actions API. Hosted on TechSoup Global's NetSquared Platform and sponsored by PayPal, Convio, TakePart, and Challenge Your World, the Change the Web Challenge offered $10,000 in prize monies for fully functional applications that connect more people with action.

The Challenge ran for 5 weeks, after which an online public vote narrowed the list of 35 submissions to 24 finalists. A team of eight expert judges then selected three winning applications consisting of an interactive map, a Firefox extension, and an iPhone application. Criteria for selecting winners was based on innovation, usability, and potential for impact. In addition to inspiring an impressive range of applications, the Change the Web Challenge also created an ongoing developer community (http://groups.google.com/group/social-actions-dev).

Along the way, Social Actions has been actively developing, with our partners, an XML schema that expands dramatically on previously-available data points specific to online and offline actions. The Open Actions XML schema (http://socialactions.com/developers/open-actions) incorporates detailed information related to each action, its anticipated impact, and affiliated organizations. As described below, the schema and other initiatives that support data standards within the online philanthropy sector are an increasingly important feature of Social Actions' work.

From Social Actions to Social Entrepreneurs

A recent opportunity allowed us to apply our experience in developing the Social Actions API to another innovative aggregation. With seed funding from the Peery Foundation (http://www.peeryfoundation.org), Social Actions is developing the Social Entrepreneur API (http://www.socialentrepreneurapi.com), the first open database of information about social entrepreneurs who have won fellowships and awards from social enterprise funders. As with the Social Actions API, these sources will provide a feed of data that is already publicly available.

The Social Entrepreneur API will help philanthropists, investors, press, and fellow entrepreneurs find social entrepreneurs based on keyword, location, cause area, population served, and a variety of other factors.

Five social entrepreneur platforms are currently participating:

- Social Edge (http://socialedge.org), a program of the Skoll Foundation (http://www.skollfoundation.org)
- ideablob (http://www.ideablob.org)
- PopTech (http://www.poptech.org)
- the Draper Richards Foundation (http://www.draperrichards.org)
- Civic Ventures (http://civicventures.org)

Social Edge will be one of the first organizations to make use of the Social Entrepreneur API in the form of a search engine on its site. As with the Social Actions API, this open dataset will be available for any website or individual to search, syndicate, republish, or use to build web applications, widgets, and search engines. We're actively facilitating the dataset's distribution by convening a group of organizations interested in using the Social Entrepreneur API as an online resource, including Dowser (http://www.dowser.org), the Fast Forward Fund (http://fastforwardfund.org), Foundation Source (http://foundationsource.org), PureProject (http://pureproject.org), and TakePart (http://www.takepart.com). These groups will test an alpha version of the Social Entrepreneur API before it launches later this summer. We'll be sharing case studies of the impact this new resource is likely to have for this broad community of social entrepreneurs.

While technically similar to the Social Actions API, the collaborative process surrounding the development of the Social Entrepreneur API is different. A small number of organizations were invited to actively participate in its conceptualization and build-out. These groups will determine as a whole when and how to invite additional participants. Importantly, participants have decided on a minimum taxonomy as well as a process for adding data or tags over time. This project mirrors the Social Actions API in its transparency: conference call transcripts, the XML schema, and other documentation are all posted to an open-to-the-public Social Entrepreneur API Google group (http://groups.google.com/group/social-entrepreneur-api).

Social Actions recently launched a consulting practice to serve foundations, companies, and nonprofits that are interested in using social media to engage more people in making a difference. Over the past several months we have worked with The Case Foundation (http://www.casefoundation.org), The Mozilla Foundation (http://mozilla.org/foundation), NABUUR (http://www.nabuur.com), and others on a range of social media projects consistent with our mission. Consulting will most likely continue to be an important source of revenue for some time. We had recognized early on that we didn't want to derive revenue from the Social Actions API, whether via fees from action sources or a commission from the traffic generated. Neither did we want to insert any kind of advertising on our search engine that would distract people from getting involved in philanthropic campaigns. We recently hosted a fundraising campaign (http://socialactions.com/2009-fundraiser) to which over 100 people contributed a total of $14,000. This support inspires us and reflects an important consistency in Social Actions' commitment to encouraging microphilanthropic campaigns.

Collaboration as Principle and Practice

Social Actions' goal is to make the Web more philanthropic, and we see an important, and often overlooked, consistency between the praxis of philanthropy and the principles that inform open source design, decision making, and management. We are building an open source infrastructure that engages a community of partners and inspires individuals to take action. Our challenges are not technological. In order for Social Actions to fulfill its mission, we need to effectively create safe places for collaboration and open dialogue at all levels within our own organization, across the sector, and among the multiple sectors that inform and impact our work.

Collaboration (working with) is often contrasted with competition (two or more striving for something only one can possess). While as concepts and practices they are quite distinct, we suggest that placing them in an either-or relationship buries opportunities to innovate through collaboration. In June 2009, Peter hosted a discussion on Social Edge (http://socialedge.org/discussions/social-entrepreneurship/collaboration-versus-competition) that specifically explored the costs and benefits of pursuing both kinds of relationships in our sector and to draw attention to "the possibility that opportunities to innovate are lost by groups too closely subscribing to the notion that competition is a good thing".

Looking Ahead

Whether via the applications that have been built for the Social Actions API, YouTube's Call to Action Overlay (http://www.citizentube.com/2009/03/youtube-nonprofit-raises-10000-for.html), or the newly launched All for Good (http://www.allforgood.org/) service and the applications that will be built for it, it is becoming easier for people to turn their good intentions into meaningful action. Two factors are driving this trend: i) the technology for collecting, distributing, and increasing access to nonprofit data is advancing; and ii) there is a cultural shift compelling technology companies, media outlets, and bloggers to use their influence to direct people to action.

As businesses seek ways to carry out their corporate philanthropy and social responsibility programs online, the infrastructure that Social Actions is developing will prove invaluable. Businesses will have a range of APIs and linked datasets that will allow customers and employees to seamlessly connect with actions that they perceive as impact generating. The open source repositories of action will reduce the costs associated in developing new and innovative programs for the private sector.

Peter Deitz is a blogger, social media consultant, and the founder of Social Actions. He is a guest blogger on Social Edge, the Stanford Social Innovation Review, and PopTech. Peter has spoken at several 2009 venues including the Nonprofit Technology Conference, the NetSquared Conference, Connecting Up Australia, Semantic Technology Conference, and My Charity Connects. Peter holds a BA in History from McGill University and an MA in History from the University of Toronto. He lives in Montreal, Quebec.

Christine Egger is a founding team member of Social Actions. She holds a master's in International Development from Michigan State University and brings 15 years' experience in for- and non-profit project management, fund development, networking and collaboration, and strategic planning to the Social Actions team. Her work focuses on the intersection of international development, philanthropy, and the complexity sciences. She lives in southeast Michigan.

"This is how the economy comes to know itself."

Gregory Norris, Director of Sylvatica

Companies are increasingly being pressured to "be green," although it is not always clear what this means. Upon closer examination, the concept of green can be seen as an emergent quality of the interactions between many companies, many chemicals, and our environment, all driven by our collective purchases. This tangled web can be better understood with the right analysis methods, software, and data, but these resources are currently scarce and expensive. Companies needing to navigate this landscape can invest their intellectual capital by encouraging the construction of an open environmental infrastructure of tools and data. This article discusses the analysis methods, software, and data that can be used to help companies, the economy, and society become greener, faster.

Understanding Green

"What we want," Jim said, "is to know if we can switch suppliers without encouraging more pollution." He was concerned how his purchasing decisions would be viewed by his customers and environmental groups. Jim uses millions of glass bottles each year, and he is very careful to maintain his brand image as a responsible company. He continued to explain his dilemma.

"We currently buy from a glass facility right here in the United States. And now we're looking at a plant in Asia, but there are a lot of reservations about the additional miles the bottles will travel to get here. We know that the Asian facility is new, and the American one is not. So our question is, 'Do the energy efficiencies of glass from this Asian plant outweigh the environmental burden of shipping them half-way around the world?' "

Jim's concerns are not unique. He wants to buy green, but there is no clear definition of what green might be. In this case, a basic heuristic that people employ to understand whether something is green has failed. One often thinks of local goods as better for the environment because of reduced shipping needs. However, Jim found himself in a position where he questioned this proposition, and went looking for a quantitative answer.

As it turned out, Jim's instinct was correct. When all the data were gathered and analyzed, it was clear that the most environmentally impactful part of the glass bottle supply chain is actually fabricating the product. Meanwhile, the burden from final shipping to the United States was minimal. These facts, combined with the energy efficiency of the Asian facility meant that, despite its long journey, the foreign glass was much 'cleaner' than the domestic option.

Life-Cycle Research

In the above anecdote, something important was glossed over: how was the glass bottle data gathered, and how was it analyzed? The comparison of the two glass bottle suppliers was made across the entire supply chain, which includes raw material extraction, transportation to the glass factory, glass making, packaging, and final delivery. Industrial activities used in each of these steps were included, such as making the cardboard box for packaging, and growing the trees to make the cardboard box. One could continue to describe how the delivery trucks were made, or the construction of the glass facility, but the point is that these studies aim to be very comprehensive. The studies are often called Life Cycle Assessments (LCAs, http://en.wikipedia.org/wiki/Life_cycle_assessment). LCA is a standardized practice described by the ISO series 14040 standard.

The building block of an LCA is a unit process which represents one node in a supply chain. Once a unit process' data are specified, it can be connected to other nodes in a series of sales and purchases. An economic network emerges that represents how goods and services are exchanged in the real world. To gather data for a unit process, one answers the following four questions:

1. What do you sell?

2. What do you buy?

3. What do you emit to air, water, land?

4. What do you extract directly from nature?

The LCA for glass making might go something like this:

1. **What do you sell?** One 120cc glass bottle.

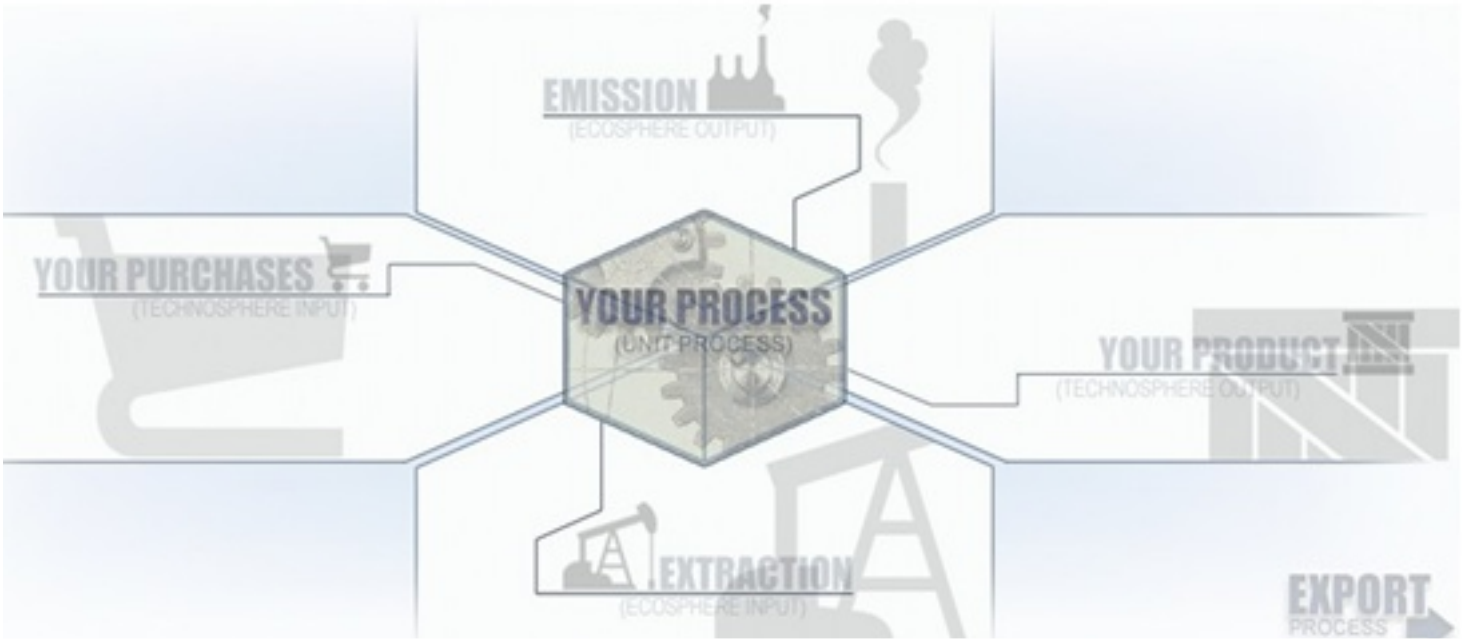
2. **To make your product, what do you buy?** 100 grams of sand, 10 grams of soda powder, 7 grams of limestone, 3 grams of dolomite, 0.5 grams of inorganic chemicals, 0.5 kWh of electricity, 0.05 MJ of heavy fuel oil, and 100 grams of waste disposal services.

3. **To make your product, what do you emit?** 6.2 grams of carbon dioxide to air, 0.5 grams of nitrogen oxides to air, 0.03 grams of sulfur dioxide to air, 0.003 grams of hydrogen chloride to air, 0.05 grams of total suspended solids to water, and 0.0001 grams of zinc to water.

4. **To make your product, what do you extract?** 1.4 cubic metres of water from the river.

For this glass-making process to connect to its sand purchase, there must be a unit process for sand. There must be unit processes for limestone, dolomite, electricity, and all the other purchases. Jim will want to connect his unit process to this glass making process as one of his many purchases.

Figure 1: The LCA Unit Process



The array of connections can quickly become dizzying as every activity in the economy needs a unit process. Where there are no company-specific unit processes, generic databases that give production recipes and emissions factors on average, such as for all coal-fired electricity plants in the US, are used. We are always looking to improve them with primary data directly from the companies.

This power to dig into one's supply chain is very valuable, transformative, and sensitive. Supply chains go on indefinitely and are interwoven into a tightly interdependent network to form our economy. When Jim purchases a glass bottle, the glass factory purchases electricity, which uses glass somewhere in its supply chain, which uses electricity, which uses glass somewhere in its supply chain, and so on, to infinity. Other exchanges, such as coal and electricity, are more closely coupled, with each successive purchase smaller than the previous. This is an infinite series in mathematical terms.

It is worthwhile to reflect on what this all means. Is every company part of every other company's supply chain? How many tiers back must we go to see this phenomenon happen? How do we assign responsibility for pollution in a world where it takes billions of parties, all pushing on each other, to make the pollution happen? Is it the seller's fault? Is it the buyer's fault? Is it the consumer's fault? Is it even worthwhile to try to parse responsibility, or should we just accept that our neighbour's pollution is also ours? Just as it takes "two to tango," it apparently takes "billions to pollute." The infinite loops in our economy tell us that we are literally all in this together.

Important Challenges in the LCA World

One can look at an economy as a big network whose behaviour we are trying to understand and moderate. LCA attempts to break apart the economy into its fundamental building blocks and to examine how they interact to produce that high-level behaviour. The discipline considers unit processes to be those fundamental blocks. We have much work to improve LCA data, software, and methodologies. We will discuss important aspects of the first two. Good data and capable software with current LCA methodologies can dramatically change how business is done, and how we live by extension.

Data Collection

The information needed to build unit processes for each economic process is all around us. Companies know what they buy and what they sell. Larger companies with facilities that contribute the most pollution to the atmosphere, surface waters, and landfills generally know the details of what they emit. What is rarer is that these data are assembled together in a way that specifies purchases, emissions, and extractions per unit of product or service produced. All the pieces to create unit processes often exist in organizations, but they are not yet coordinated across departments to create a product-oriented view.

One major challenge for LCA is to gather these data and make them available to downstream supply-chain actors. While this is not a difficult process in theory, it takes quite a bit of effort because information systems and company departments are not set up to analyze process operations this way. Collection of the environmental data invariably falls into the lap of operations or environmental engineers, who have other priorities, and are not accustomed to dealing with customer requests. The real difficulties often lie in the gathering of purchasing information where the main barrier is confidentiality.

Companies do not want to disclose what they are buying to fill their product orders. It's a very sensitive topic. Overall, today's LCA data collection is onerous, expensive, time-consuming, and difficult to navigate because the data are controlled by different people, with varying levels of trust for outside LCA researchers and consultants.

Data Sharing & Tools

The calculation of an LCA doesn't require access to all the unit processes in a supply chain. A unit process can be aggregated into a system process. A system process blends the emissions from all purchases with a unit process' direct emissions, and presents it as if it were the pollution of the unit process itself. In other words, a system process obscures the purchases in a supply chain, while maintaining the vital environmental emissions information. Companies may:

- create a unit process
- connect to their supplier's system processes
- aggregate their own unit-process into a system process
- make their system process available to their customers

The cycle can continue throughout the entire supply chain, enabling all actors to have supplier-specific life-cycle information without direct disclosure of sensitive purchasing and process information. One very important consideration, which is inevitably the first objection raised, is how exactly are unit process data verified to ensure their accuracy and reliability? Some have envisioned a profession similar in nature to financial auditing, in combination with algorithmic checks that look for anomalies from year to year. Some have advocated for a scarlet letter system that identifies past cheaters. Certainly, this problem is not yet settled.

Imagine a software application that would enable this type of interaction. It would allow companies to create a unit process for their product, and to connect to real system process data of suppliers. The company would also specify their emissions and extractions per product unit. They could then prioritize the activities in their supply chain that are causing the most environmental impact. Is it the steel purchase, the glass purchase, or one's own natural gas combustion? Once environmental priorities are set, various alternatives can be compared. When the company has completed their unit process, and it has been validated, they have the option to publish it, thereby making it available for customers to use in their calculations. The cycle continues with participation from each producer, until our entire economy is connected.

The vision of the Earthster (http://www.earthster.org/) project is to create an open source application that enables confidential LCA calculation and linkage. The project is still in early phases, but it is safe to say that an application of this nature will emerge in the near future. Such software must ensure good data quality, confidentiality, and scalability to operate in a world where billions of economic transactions happen every day. This software should not be just one application. It should be many, operating in an ecosystem that has been built to encourage low-cost of participation and innovation. The ecosystem must be open, and at least some of these applications should also be open.

The Need for Open Source LCA Tools and Data

LCA is a pragmatic discipline, grounded

in the notion that its use by individual actors will simultaneously benefit self-interested producers and the public at large. Companies can employ it to find pollution and inefficiencies in their supply chains during supplier selection, or it can guide their product development efforts. Green supplier selection or product design is a boon to the public both in terms of reduced pollution and as a source of inspiration to product designers.

To date, the LCA data and software model has been to develop centralized datasets and encapsulated desktop applications. Open LCA (http://www.openlca.org), an open source java application accessible through Eclipse, is one notable project. Such efforts have enabled the LCA profession to blossom as software greatly simplifies the gargantuan task of maintaining millions of links between unit processes in the economy, and the datasets are a reliable set of average information on which good LCA studies can be conducted. The database initiatives painstakingly map out the unit processes of many agricultural, industrial, and service processes in the economy at significant cost. Great care is taken within the LCA database initiatives to guarantee data quality and consistency, which is no small feat.

There is room, and perhaps a great need, for another kind of data and software model in the LCA world. This is a crowdsourcing approach where the economy maps itself. Companies can move from using generic data to company-specific information about the products they are actually buying. There would be no centralized intermediary to slow or alter the flow of this information, just as there is no centralized clearing-house for product pricing in our current economy.

The costs of gathering and validating the data are huge, but this challenge can be tackled with widespread collaboration. A framework is needed to coordinate these efforts while also encouraging experimentation. The business case for such a framework is not clear, but our society's need for it is. One can only hope we are able to build a common vision before fragmentation and lock-in from private vendors takes place. Proprietary applications will certainly contribute greatly to increased environmental transparency in supply chains, but it is important that at least certain pieces of this framework remain open source.

It is certain that the world needs an open source, flexible data exchange standard for LCA data. This task is complicated by virtue of LCA data never standing on its own, but rather always being part of a larger network. When a process leaves a software installation and is imported into another one, the receiving software must determine exactly where the process can be plugged into the existing network. This is a complex task and best practices are constantly evolving. An open and flexible data format should enable software applications the leeway to experiment with new ideas, without breaking the validity of the format or the ability to exchange basic information with other applications. There are currently two major LCA data formats, EcoSpold (http://www.ecoinvent.org/we-about-us/ecospold-data-format) and ELCD (http://lca.jrc.ec.europa.eu/lcainfohub/datasetArea.vm). EcoSpold is controlled by the Swiss Non-Profit, EcoInvent, and ELCD is the data format of the European Life Cycle Data System. While these formats may not be technically open source, they may be able to serve many of the needs listed above. It remains to be seen if they can offer the necessary flexibility.

It is also important that there is at least one open source LCA-browser. This browser is software that enables a user to access the datasets, explore how

processes are connected to each other, and identify major environmental hotspots in the supply chain. It allows users to create their own unit processes and publish them to the LCA network. The primary argument for at least one open source browser is to provide a basic platform on which individuals or small teams can innovate. The browser analogy can be continued by calling these small innovations add-ons or plug-ins. LCA is a quickly evolving field, and an open source core would encourage small-scale experimentation and learning. Major breakthroughs could then be integrated into the core open source browser.

As with the software, it is beneficial to have some open source datasets. High quality generic LCA data will always be required to fill gaps where companies are not reporting, and this is especially true for the next 10-20 years when those gaps will be significant. Collaboration on open source datasets can be a key component to improve generic LCA datasets. Transparent and open datasets may enable faster quality improvement because of their transparency. Industries and companies that perform poorly in these datasets are able to understand why and then fix the dataset, fix their production processes, or both. Transparency enables non-profits and research institutions to spot weak assumptions and offer better information.

One of the largest barriers to wide-spread implementation of LCA has been its high costs. It is expensive to collect the data and to design analysis tools. These costs are not operating costs, but rather startup costs. Once the data are collected, they can be re-used by everyone in the world at a small marginal cost.

In this sense, the LCA community faces the same situation as many software makers and the music industry. However, the world needs LCA to become ubiquitous to address our current environmental imperative. As we build the LCA infrastructure, it can be extended to other aspects. Once we have detailed LCA models, we can go beyond environmental pollution and ask human rights questions of supply chains.

It is necessary to lower the costs of LCA for smaller enterprises and to show its value to larger corporations. This is achievable, provided we put the infrastructure in place, so that the world achieves maximum benefit. All interested parties must collectively work to foster an open ecosystem that promotes participation, innovation and the transformation of our economy by making it self-aware.

Summary

There is an increasing need for tools and data to address the environmental imperative. LCA is an important contributor due to its ability to cope with realistic interconnections of economic actors and to identify leverage points for environmental improvement. Needed collaboration is emerging to enable real-time, decentralized sharing of product environmental information. We must foster trust, participation, and experimentation to green our economy as quickly as possible, so our economic lives can reflect our values as a people.

The title of this article is derived from a quotation fragment by Pascal Lesage, Director of Sylvatica in Montréal.

Evan Andrews is an analyst at Sylvatica (http://www.sylvatica.com), a Life Cycle Assessment consulting firm and research group. Sylvatica's mission is to propose creative solutions to improve, and eventually make positive, the environmental and social impacts of our actions. It administers the Earthster Project as one of its many initiatives. Evan is based in Montréal, Québec.

"For my generation the great innovation was the course team. For the next I suspect that it will be Open Educational Resources."

Sir John Daniel

The education community has been at the forefront in envisioning and conceptualizing infrastructures intended for utilizing and sharing digital content or resources. However, this community has faced challenges in making these visions a reality. We begin by describing a relatively early attempt at creating an economy for sharing educational resources, referred to as learning objects. We then discuss two approaches to opening up educational contents to the world under the auspices of the more recent Open Educational Resources (OER, http://en.wikipedia.org/wiki/Open_educational_resources) movement. One of these approaches has focused on creating open resources from scratch, utilizing Wiki content development and management technologies in the wake of the phenomenal success of Wikipedia. A second approach is represented by developments in Open Courseware. Following the example of MIT's Open Courseware (OCW, http://ocw.mit.edu), this approach has more recently been adopted by many other educational institutions under the OCW Consortium (http://ocwconsortium.org). We conclude by making the case that this second approach may represent the most promising of recent developments in the adaptation of open source and open content to educational practices and technologies.

Learning Objects

The term "learning object" has been defined in a number of different but broadly congruent ways. It is significant that each definition highlights modularity as a technological and design attribute for the object and its content, emphasizing the "self-contained," "building block" or "object-oriented" nature of the technology. Learning objects were not typically associated with whole courses, but were seen as optimally being comprised of smaller modules, units or course subcomponents. The use of terms like "modular," "digital" and "object oriented" testify to a broader emphasis on the technological solutions and standards evident in many learning object projects and publications. Technically-based interchangeability and interconnectability implied by the term "object oriented" has been an important issue in learning objects discussions over the years.

The term learning object was first popularized by Wayne Hodgins in 1994 (http://journals.tdl.org/jodi/article/view/89/88). The relatively early date of this coinage is significant: it is roughly simultaneous with the popular emergence of the Web itself. This means that the development of practical and technical conventions, technological solutions and standards for the interchangeability of these objects were the first of their kind for any type of distributed content on the Web. These path-breaking standards include the IEEE Learning Object Metadata Specification (http://ltsc.ieee.org/wg12/) and the IMS Content Packaging Specification (http://www.imsglobal.org/content/packaging/). The emergence of these standards specifically for education provides an important example of educational technologists being at the leading edge of developments in technology. The early development of these bleeding edge educational specifications in some ways outpaced more general legal, practical and technological developments. They consequently may have ended up creating rather than resolving problems for educators.

Some of the most ambitious visions for learning objects saw complex, interactive educational resources, whether informational or interactive and software-based,

as being combined together in the context of a powerful but flexible "component architecture". "We argue that stand-alone applications are incompatible with typical production, distribution, and usage patterns for educational software. We aim to convince the reader that emerging industry-standard component software architectures "[will allow] a comprehensive learning works [to] emerge [on the basis of] contributions from many distributed innovators" (http://www-jime.open.ac.uk/98/6/).

Some saw such emerging industry-standard component architectures as enabling not only new levels of technical interoperation and ease-of-use, but also facilitating the development of new communities, practices and even economies. In an industry white paper entitled "Elusive Vision: Challenges Impeding the Learning Object Economy" (http://download.macromedia.com/pub/solutions/downloads/elearning/elusive_vision.pdf), Lawrence Johnson describes the basis for such an economy: "Commercial exchanges are the heart and soul of any market economy, and in the commercial market for learning objects, end users and aggregators purchase content under specific licenses that allow them to use the objects in clearly defined ways. This arena includes large traditional publishers who want to repurpose their content as learning objects and training companies eager to move into e-learning. Also appearing are a crop of smaller new entrants who publish learning objects as their core business. This market has some special challenges, and many issues related to licensing remain to be sorted out."

Such an economy was not visualized as being open in the sense of open source or OER. Johnson's final observation that "many issues related to licensing remain to be sorted out" provides a clear indication as to why such an economy did not develop, and also why openness has since become much more important in discussing resources for learning and education. The issues related to licensing have ultimately proven virtually impossible to "sort out." At the time of Johnson's statement, many saw digital rights management (DRM) as a technical answer to questions of licensing, particularly in the contexts where modular resources would be recombined and repurposed in complex ways. It would allow uses of digital bits of content to be prescribed and controlled in great detail. DRM technologies would grant or prohibit forms of access and use of a learning object according to a legal license. These technologies have been successfully challenged and undermined in the world of popular music and video, and they have met with even less success in the world of education. This is one among many reasons why the widespread adoption of learning objects, either on a commercial or a more open basis, has not yet occurred. The innovative approach taken by the OER movement to this challenge constitutes one of its most important characteristics.

Open Educational Resources

OER is a term first adopted at a 2002 UNESCO Forum on the Impact of Open Courseware for Higher Education in Developing Countries. The phrase was defined as "the open provision of educational resources, enabled by information and communication technologies, for consultation, use and adaptation by a community of users for non-commercial purposes" (http://unesdoc.unesco.org/images/0012/001285/128515e.pdf). This definition and its emphasis on open availability and non-commercial use remains central in the way OER are understood today. UNESCO's far-reaching humanitarian goals are still very much relevant to the use of the term OER today.

Creating OER: The Wiki Model

The first OER model we discuss is the creation of open educational content from scratch in online Wiki environments specially designed for the organization and collaborative development of such resources. The Wiki-based approach is primarily associated with the Wikiversity (http://en.wikiversity.org) and WikiEducator (http://www.wikieducator.org) initiatives. Both were founded in 2006 and share many points of commonality in terms of process, form and content.

Wikiversity was launched with the aim of "...empower[ing] people to achieve their educational goals using resources produced by the free culture movement. The goal...is to create a community of people who support each other in their educational endeavors." WikiEducator has set itself a slightly more ambitious and specific task: To work "collaboratively with the Free Culture Movement towards a free version of the education curriculum by 2015." It is significant that both make clear reference to the free culture movement associated primarily with Creative Commons (http://creativecommons.org) and other alternatives to common copyright restrictions. WikiEducator departs from Wikiversity in emphasizing the development of contents for formal education.

WikiEducator and Wikiversity are not limited to addressing post-secondary learning needs, but are designed to serve many educational levels. Both provide separate portals for primary, secondary, tertiary, and other categories of education. Wikiversity's portals offer a number of resources such as courses, discussions, essays, handouts, lesson plans, presentations, reading groups, study guides and syllabi. WikiEducator's portals contain a variety of resources with varying forms of organization and content types.

The ambitious range of resources, services and educational forms and levels encompassed by Wikiversity and WikiEducator is evident in their recent articulations of their surprisingly congruent primary priorities and goals:

- build capacity in the use of Mediawiki and related free software technologies for mass-collaboration in the authoring of free content (WikiEducator, 2009)

- create and host a range of free content, multilingual learning materials and resources, for all age groups in all languages (Wikiversity, 2006)

WikiEducator is sponsored, in part, by the Commonwealth of Learning (http://www.col.org), "an intergovernmental organisation created by Commonwealth Heads of Government to encourage the development and sharing of open and distance education knowledge, resources and technologies." WikiEducator places significant emphasis on international development. Wikiversity is a brainchild of the Wikimedia Foundation (http://wikimediafoundation.org), which is also responsible for Wikipedia. Wikiversity and Wikipedia currently share eight sister projects ranging from Wikimedia Commons (http://commons.wikimedia.org) to Wikispecies (http://species.wikimedia.org). Wikiversity aims for a general and widespread impact, covering both formal and informal types of education, for learners in wealthy as well as developing countries.

Wikiversity and especially WikiEducator sponsor workshops to build capacity and enable volunteers to create content using the Mediawiki (http://www.mediawiki.org) content development and management software. At the time of writing, WikiEducator has delivered over 100 workshops to over 2,000 participants. Wikiversity boasts that it has over "10,537 learning resources and growing".

The Open Courseware Model and MIT

The second approach we discuss is the conversion of existing classroom course content to make it freely available on the Web. MIT's OCW initiative focuses on the conversion of conventional classroom resources. Announced in 2001, the project's goals were originally described in the press as follows: "[MIT] announced plans to post on the Internet materials for nearly all of its courses. Access to the materials, which will include lecture notes, course outlines, reading lists, and assignments, will be open to the public and free of charge. The information posted could be used as reference material, as a source for curriculum development, or as a foundation for independent study" (http://aaup.org/AAUP/pubsres/academe/2001/JA/NB/MIT.htm).

This approach to OER has met with considerable success. MIT met its own ambitious goal of posting "virtually all" of its courses online by 2007. The MIT project is also noteworthy for its emphasis on MIT's own institutional products and for being one of the few early, high-profile online initiatives announced by a campus-based institution to survive to the present day. The project effectively pioneered the notion of free access to course materials, and popularized the term "open courseware."

The OCW Consortium, founded in 2005, takes MIT's OCW approach to a consortial level, bringing together MIT's courses with those of many other universities internationally. At the time of writing, this consortium includes over 200 members and affiliates and has brought together about 10,000 courses. The consortium defines its principle goals as follows:

- extend the reach and impact of open courseware by encouraging the adoption and adaptation of open educational materials around the world
- foster the development of additional open courseware projects
- ensure the long-term sustainability of open courseware projects by identifying ways to improve effectiveness and reduce costs

This last goal is of no small importance to open approaches to educational content, courses and other resources. It forms the focus of the concluding section of this paper.

An Open Question: Sustainability

Sustainability, the capacity of an initiative to outlive its initial startup phase and the associated short-term project funding, is a major concern for OER projects. Both wiki-based resource sets and collections of courseware must find long-term support or revenue. They must develop their particular work from the status of a project to become a program, organization or consortium. It is disquieting to read in a recent report on OER that "the majority of OER development" are generally still being "undertaken on a project basis" (http://oerwiki.iiep-unesco.org/images/4/46/OER_Way_Forward.pdf).

OER activities, specifically when they follow the OCW model, present a relatively clear alternative to project funding: the financial support of the educational institutions with which the courses are associated. Reasons for providing ongoing funding can be compelling for an institution. A number of motivating factors are outlined in the findings of a 2005 Program Evaluation Findings Report (http://ocw.mit.edu/ans7870/global/05_Prog_Eval_Report_Final.pdf) produced as a part of MIT's OCW project:

1. The majority of the use of MIT courses is for self-directed, informal learning: namely, to "improve" or "enhance

personal knowledge" or to "explore areas outside [one's] professional field". In other words, the majority of materials use occurs outside of institutional settings. This helps to explain a contradiction apparent in the MIT initiative: it is educationally valuable but does not detract from the educational value of the face-to-face activities on which the collected content is based. The informal users of this material, generally located outside of North America, would not be potential on-campus students or "customers" of the institution generating the material.

2. A second finding is connected to the relationship of the project to MIT itself as an institution. It provides clear evidence of multiple areas of significant benefit accruing to MIT from the OCW project, and provides the strongest motivating factor for long-term local support. The report states that "OCW use is centered on subjects for which MIT is a recognized leader," with areas in technology and science accounting for 62% of traffic. Majorities of students and faculty at MIT use the site to support their study and teaching, and 32% of faculty say that putting materials online has improved their teaching.

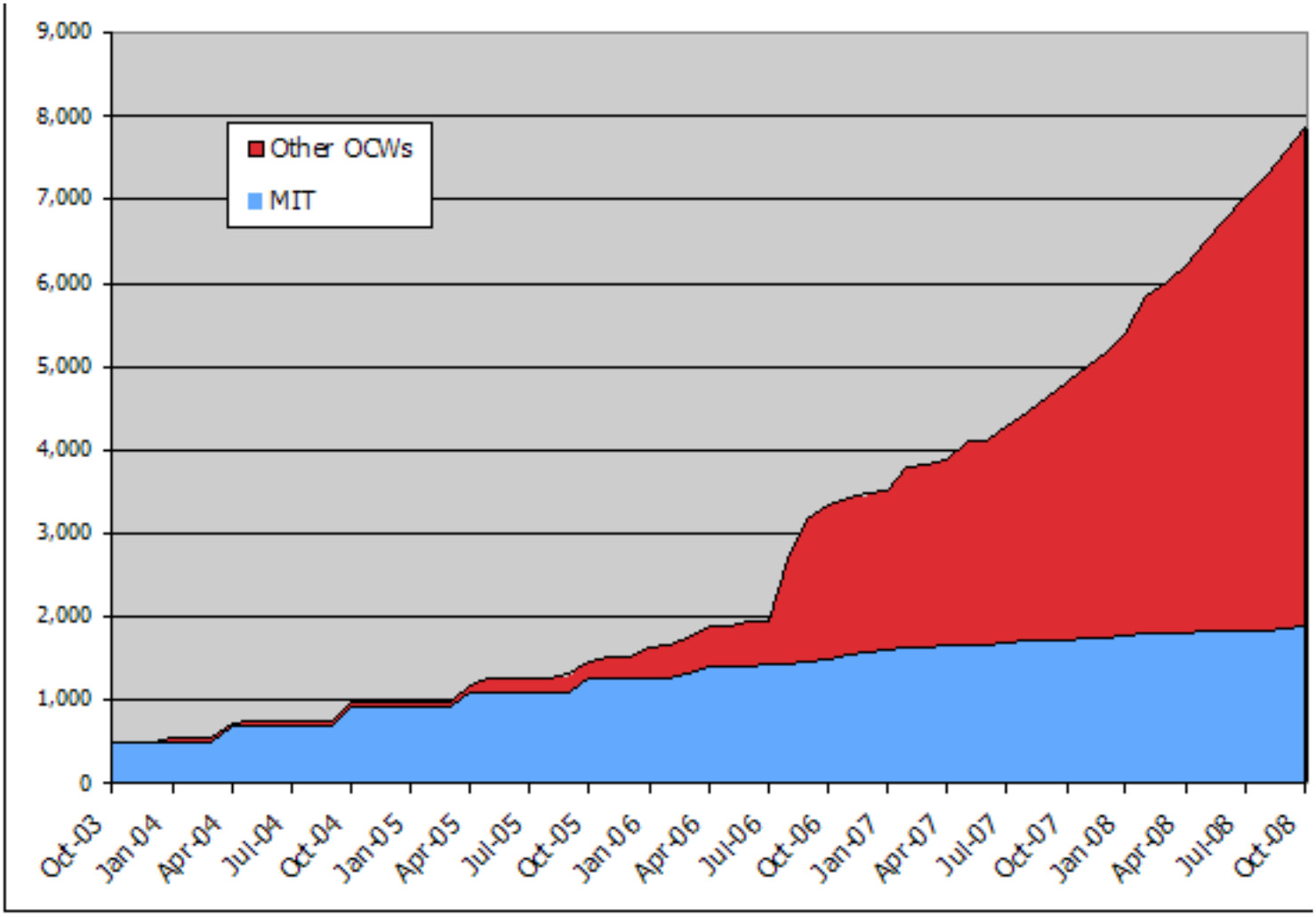
Finally, the role of the project in student recruitment is significant: 16% of student users employ the MIT courses to "plan a course of study," and "35 percent of freshmen who were aware of OCW prior to deciding to attend MIT indicate the site was a significant or very significant influence on their choice of school" (http://www.ed.gov/about/bdscomm/list/hiedfuture/3rd-meeting/wiley.pdf). Significantly, this percentage of students more than quadrupled from the year before.

Commenting on this rapidly growing awareness of student recruits, David Wiley presents a conclusion that may be of the utmost significance for OER: "The time will come when an Open CourseWare or similar collection of open access educational materials will be as fully expected from every higher education institution as an informational website is now" (http://www.ed.gov/about/bdscomm/list/hiedfuture/3rd-meeting/wiley.pdf).

Conclusion

Enlightened institutional self-interest is one of the most powerful drivers for the sustainability of OCW initiatives and for OER more generally. Wiley makes the case in connection with institutional service and recruitment, but MIT has benefited in many other ways from its early investment in OCW. Although MIT is able to leverage an already existing global reputation and first-mover advantage, many benefits would also apply to smaller institutions. These include student recruitment, the potential for improving teaching and better supporting learning, and viral marketing of the quality of teaching and learning in areas of strategic institutional interest. Those following in MIT's footsteps enjoy the advantage that effective licensing, consortia and growing awareness are all in place. They need not risk financial and cultural capital on creating yet another collection or repository. Instead, they can invest in the quality and accessibility of their course offerings. This is enabled through the OCW Consortium, which combines and centralizes course offerings to create "a broad and deep body of open educational content using a shared model". The OCW Consortium presents a relatively low barrier to entry and only asks of its members a contribution of 10 courses to its growing collection. This low barrier to entry, as well as the expanding number of reputable member institutions, has resulted in the kind of exponential growth shown in Figure 1 (http://tinyurl.com/lkvzaz):

Figure 1: Open Courseware Production in the Open Courseware Consortium



The hope is that the examples, evidence and arguments of the kind provided in the MIT report and the above graph will lead to action and investment whose effects ultimately extend well beyond present institutional interests. The point, as Wiley explains, is that "this strategy of openness" holds out the promise of "catalyzing further innovations". Innovations in practice, community and policy have the potential of fomenting the gradual, cultural sea change that is needed for the success of OCW and OER models of whatever kind.

This paper was written with the support of the BC Centre for Open Learning.

Dr. Norm Friesen is Canada Research Chair in E-Learning Practices at Thompson Rivers University in Kamloops, BC, Canada. Dr. Friesen has been developing and studying Web technologies in educational contexts since 1995, is the author of several editions of books on the effective use of instructional software, is co-editor of Phenomenology & Practice, and is a member of the Canadian delegation to the ISO subcommittee for Learning, Education and Training. He previously worked as a Postdoctoral Fellow at the School of Communication at Simon Fraser University, and as an adjunct or visiting faculty at Athabasca University, at the University of Toronto, the Technical University of Dresden and the University of Innsbruck. He holds Master's degrees from the University of Alberta and Johns Hopkins University, and a PhD in Education from the University of Alberta.

"Open source better prepares students for the business world by exposing them to real-world problems and encouraging learning through the completion of real tasks. Open source amplifies a "hands-on" approach to learning by connecting students to a community of users in an effort to solve problems."

Jim Whitehurst, CEO of Red Hat

A healthy community is the lifeblood of any open source project. Many open source contributors first get involved while they are students, but this is almost always on their own time. At Seneca College (http://www.senecac.on.ca) we have developed an approach to sustainably involving students in open source communities that has proven successful in a course setting.

This paper outlines Seneca's approach and discusses the results that have been obtained with it. We examine the key factors for successful student integration into open source communities and steps that educational institutions and open source projects can each take to improve student involvement.

Barriers to Teaching Open Source Development

To effectively teach open source, it's necessary to move each student into the role of contributor. This appears straightforward, but ultimately proves to be an enormous challenge because: i) open source is as much a social movement as a technical one; and ii) many open source practices are the exact opposite of traditional development practices.

Many attempts to involve students in open source within a course have failed because everyone is overwhelmed:

1. The students, because they're suddenly facing an established codebase several orders of magnitude larger than any they have previously encountered in their courses, a community culture that they do not understand, and principles and ideals which are the opposite of what they've learned in other courses. For example, students are taught that answers and solutions should not be openly shared on the Web (http://www.wikinomics.com/blog/index.php/2008/03/12/the-ryerson-facebook-dilemma/), that building on other's work by pasting it into your own is academically dishonest, and that it's wrong to deeply collaborate with peers on individual projects.

2. The professor and institution, because they're dealing with a continuously-changing, amorphous environment.

3. The open source project, because it is very difficult to deal with a sudden influx of students who tie up other contributors' time with questions and yet are unlikely to become long-term participants.

Distinctive Qualities of Open Source Development

In order to develop an effective approach to open source development, it's important to understand the qualities which make it unique:

1. Open source development is based around communities. These are generally much larger and more geographically diverse than closed-source development teams. They are enabled and empowered by the Web, leading to an increased focus on communication tools and internationalization and localization issues. Social issues become significant, and there is a productive tension between the need to maintain group discipline for coherence and the possibility of provoking a fork.

Often, the culture of the community is not the culture of any particular member, but a synthetic intermediate culture.

2. The codebases managed by the larger communities range up to millions of lines in size and can date back many years or even decades. They often use tools and languages that are different from those taught in post secondary institutions, or employ common languages in unexpected ways. An example is using custom APIs that dwarf the language in which they are written, such as Mozilla's XPCOM (http://mozilla.org/projects/xpcom) and NSPR (http://mozilla.org/projects/nspr). These codebases require specialized, heavy-duty tools such as bug tracking systems, code search tools, version control systems, automated (and sometimes multi-platform) build and test farms and related waterfall and alert systems, toolchains for compiling and packaging each of the source languages used in the project, and release and distribution systems. Smaller open source projects which do not maintain their own infrastructure use some subset of these tools through a SourceForge account (http://sourceforge.net), Fedora Trac instance (https://fedorahosted.org/web/), or other mechanism.

3. Most open source systems have an organic architecture. Since it's impossible to anticipate the eventual interests and use-cases of the community at the inception of a project, the project requirements and development direction change over time. Although the lack of top-down design can be a disadvantage, the flexible, modular, and extensible architecture that often results has many benefits.

Each of these distinctive qualities presents a challenge to a traditional lecture-and-assignment or lecture-and-lab format course, but can be a strength in a community-immersed, project-oriented course. Carefully applied, these strengths can be used to overcome the barriers identified above.

Preparing to Teach Open Source

A prerequisite for teaching open source effectively is a professor who has one foot firmly planted in an open source community and the other in the educational world. To turn students into contributors, you need a dedicated conduit and liaison who can introduce students to the right people within the open source community.

On the academic side, the professor needs to connect with students on a personal level and to be aware of and able to navigate within the learning and administrative context of the educational institution. On the open source side, the professor must have deep contacts within the community, understand the community culture, and know what matters to the community so that projects selected for the students have traction. The professor must effectively use the community's tools and know when to use IRC, bugzilla, and email communications. The faculty member must adhere to open source principles and use the community's products in a production environment in order to have credibility.

The size of most large open source codebases prevents any one person from effectively knowing the entire codebase in detail, a problem that is compounded when multiple languages, layers, or major components are involved. One must move beyond being overwhelmed and become effective at searching, navigating, and reading code. The professor must demonstrate how to cope and show the students how to use community resources and contacts to find answers to questions. There is no textbook for this; it is behaviour that must be modeled.

Select an Open Source Community

An effective open source course requires the support of a large open source project. Project selection usually involves the faculty member(s) who will be teaching the course.

The open source community selected must have a sufficiently large scope to provide opportunities for many different types and levels of involvement. Its products must have many angles and components, so students can innovate in corners that aren't being touched by the main developers.

The reasons for selecting a larger community are straightforward:

1. A large community can absorb a large number of students spread across the various components and sub-projects within the community. This enables students with a broad range of interests and skills to get involved in a way that interests them. It also spreads student contact across the community, preventing overload of the existing contributors. Working within a single community provides a level of coherence for class discussions and for planning labs and lectures.

2. The project's infrastructure has usually scaled to the point where it will readily support the extra contributors.

3. Large projects tend to have broad industry support, opening up possibilities for spin-off research projects and broadening the value of the students' experience.

The key to project selection is to select something so big that the professor cannot directly manage the students and they are forced to interact with the community in order to succeed.

Select Potential Student Projects

Open source communities know what is interesting and valuable to them and are in the best position to suggest potential student projects. They're not always able to verbalize these projects, so the professor may need to suggest good ideas, but the community will recognize the value of ideas as they are proposed.

Some of the best project ideas are ones that existing community members would like to pursue, but can't due to a lack of available time or appropriate hardware. Project ideas should not be critical issues that directly affect release timelines or major community goals, but they may be of significant strategic value to the community. Each person proposing a project idea should be willing to be a resource contact for that project.

Potential projects can include a wide range of activities: feature development, bug fixing, performance testing, writing test cases, benchmarking, documenting, packaging, and developing or enhancing infrastructure tools.

The projects must then be screened for viability within the course context:

1. Are they the right size for the course? This does not mean that the project should be fully completed during the course. We look for projects that are not likely to be completed but which can be developed to a usable state in three months.

2. Are the necessary hardware and software resources available?

3. Is the level of expertise required appropriate for the type of student who will be taking the course? Ideally, each project should make the student reach high, but be neither stratospherically difficult nor trivially easy.

Prepare the Infrastructure

Each open source community has its own set of tools, and it's crucial that students use those tools so that community members can share with, guide, and encourage the new contributors. Existing community mailing lists, wikis, IRC channels, version control systems, and build infrastructure should be used by the students as they would by any other contributor.

There's a certain amount of additional infrastructure needed to support an open source course, including:

1. A course wiki for schedules, learning materials, labs, project status information, and student details. If this wiki is compatible with the community's wiki, it will be easier for the community to contribute to learning materials.

2. An IRC channel set up in parallel to the community's developer channel(s), on the same network or server. These provide a safe place for students to ask questions which may provoke flaming in developer channels.

3. A planet to aggregate student blogs so community members can stay up-to-date. This should be separate from the community's main planet because some of the material will be course-specific.

4. Server farms and/or development workstations to ensure that the students have access to all relevant hardware and operating system platforms.

Teaching the Course

We start our open source courses by briefly teaching the history and philosophy of open source. We don't spend a lot of time on this topic because the philosophy will be explained and modeled in every aspect of the course.

Since open source is by its very nature open, we get students communicating immediately. They are required to establish a blog and submit a feed to the course planet. Almost all work is submitted by blogging, and students are expected to enter comments and to blog counterpoints to their colleagues' postings.

All course materials and labs are placed on the course wiki, and both students and community members are encouraged to expand, correct, and improve the material. These resources and the knowledge they represent grow over time and are not discarded at the end of each semester. This body of knowledge eventually becomes valuable to the entire community. Students are also required to get onto IRC. Since the main developers channels can be daunting, students are initially encouraged to lurk in those channels while communicating with classmates and faculty on the student channel.

At the start of the course, students begin reviewing the potential project list and are required to select a project by the third week. As part of the selection process, students will often use IRC or email to contact the community member who proposed a project that they are interested in. This is the first direct contact between the student and a community member, and since the student is expressing interest in something that the member proposed, the contact is usually welcome. It is critical that students choose projects that are important to the community and attract community support, so we prohibit them from proposing their own projects. Students often find it intimidating to select from the potential project list and the professor will often need to serve as a guide during project selection.

We prefer that each student select an individual project, with some rare two-person groups where warranted by the project's scope. Larger groups are almost always less successful. Students need to collaborate in the community instead of doing traditional, inward-focused academic group work. Students claim a specific project from the potential project list by moving it to the active project list and creating a project page within the course wiki.

Tools and Methodologies

Each community has a unique build process. This is often the first non-trivial, cross-platform build that students have encountered, so it's a significant learning experience that has a gratifying built-in reward. Students often go to great lengths testing different build options and approaches. Students also learn how to run multiple versions of the software for production and test purposes.

One of the challenges with building is finding an appropriate place to build, since many of the laptop computer models favoured by students have low CPU or memory, while student accounts on lab systems may not have sufficient disk space or student storage may be shared over a congested institutional network. Possible solutions include using external flash or disk drives with lab systems, or providing remote access to build systems.

As the students start work on their project, the course topics and labs teach the tools and methodologies used within the community. In most cases, the bug or issue tracking system drives the development, feature request, debugging, and review processes, providing an effective starting point. It's best that student projects have a bug/issue within the community tracking system, so students must either take on an existing bug or create a bug/issue for each project.

One useful exercise at this stage is to have the students "shadow" an active developer. On Bugzilla, a student can do this by entering that developer's e-mail address in their watch list (http://www.bugzilla.org/docs/3.0/html/userpreferences.html#emailpreferences), which forwards the student a copy of all bugmail sent to the developer. After coming to grips with the e-mail volume, students learn a lot about the lifecycle of a bug through this process.

Next, students need to learn how to: i) use code search tools such as LXR (http://lxr.linux.no), MXR (http://mxr.mozilla.org) and OpenGrok (http://opensolaris.org/os/project/opengrok/); ii) skim code; and iii) know who to talk to about specific pieces of code, including module and package owners and community experts. By working shoulder-to-shoulder with community members, particularly on IRC, they learn the ins-and-outs of the development process including productivity shortcuts and best practices. The professor can keep his finger on the pulse of the activity through IRC, guiding students when they get off track and connecting them with appropriate community members as challenges arise.

Students are expected to blog about their experiences on a regular basis, and all of the students and the community benefit from this shared knowledge. At the same time, differences between the student projects prevents one student from riding entirely on the coattails of other students.

Guest lectures by community developers have a powerful impact on students. Meeting a coding legend on IRC is great, but talking face-to-face and seeing a demonstration or hearing first-hand about the direction the software is headed has exceptional value. We film these meetings and share the talks under open content licenses, making them available to people around the world.

We've been surprised at the number of video views and by who is viewing them. We've found that new Mozilla employees often read our wiki and view the videos to help them come up to speed on the Mozilla codebase.

Releases and Contributions

Following the "release early, release often" mantra, students are required to make releases on a predetermined schedule. For the first open source course, three releases from 0.1 to 0.3 are required, and for the follow-on course, six biweekly releases from 0.4 to 1.0 are required.

We define the 0.3 release as "usable, even if not polished", reflecting the fact that a lot of open source software is used in production even before it reaches a 1.0 state. This means that the 0.3 release should be properly packaged, stable, and have basic documentation. It may be missing features, UI elegance, and comprehensive user documentation. The slower release rate in the first course is due to the initial learning curve and the fact that setting up a project and preparing an initial solution are time-consuming tasks.

As active members of an open source community, students are required to contribute to other open source projects, either those of other students or other members within the community. This contribution, which can take the form of code, test results, test cases, documentation, artwork, sample data files, or anything else useful to the project, accounts for a significant portion of the student's mark. Each project is expected to acknowledge external contributions on their wiki project page, and to welcome and actively solicit contributions from other students and community members. This requires that they make contribution easy, by producing quality code, making it available in convenient forms, and by explicitly blogging about what kind of contributions would be appreciated.

Students are often surprised to find community members contributing to their projects, but that is part of the authentic open source experience. It's important not to choke off collaboration for the sake of traditional academics.

In order to receive credit for contribution, students must blog about their contributions to other projects. At first this seems immodest to students, but the straight-facts reporting of work accomplished is a normal part of open development.

Seneca's Experience

Seneca College has been involved with open source for over 15 years, starting with Yggdrasil Linux (http://en.wikipedia.org/wiki/Yggdrasil_Linux) installations in 1992. In 1999, we started a one-year intensive Linux system administration graduate program. In 2001, we introduced the Matrix server cluster and desktop installation, converting all of the hundreds of lab systems to a dual-boot configuration, which enabled us to teach the Linux platform and GNU development toolchain to students right from their first day at the college. In addition, a number of college faculty members released small open source software packages, including Nled (http://cdot.senecac.on.ca/software/nled/), VNC# (http://cdot.senecac.on.ca/projects/vncsharp/), and EZED (http://cdot.senecac.on.ca/software/ezed/).

In 2002, John Selmys started the annual Seneca Free Software and Open Source Symposium (http://fsoss.senecac.on.ca), which has since grown to a two-day event attracting participants from across North America.

In 2005, an industry-sponsored research project on advanced input devices created the need to modify a complex application. The lead researcher on this project, David Humphrey, contacted Mozilla to discuss the possibility of modifying Firefox. This contact led to a deep relationship between Mozilla and Seneca which outlasted that research project and led to the eventual development of the open source teaching model described here.

Our Open Source Development course implemented this model within the Mozilla community. David subsequently developed the Real World Mozilla seminar, which packs that course into an intensive one-week format, and a continuation course was eventually added to enable students to continue development on their open source projects and take them to a fully polished 1.0 release with faculty support.

Failures and Successes

The unpredictable nature of working within a functioning open source community poses peculiar challenges. We've had situations where a developer appears unexpectedly and posts a patch that fully completes a student's half-done project. Sometime students encounter reviewers who can't be bothered to do a review, stalling a student's work for weeks at a time, and some module and package owners have a complete lack of interest in the students' work. We've also had students drop the ball on high-profile work, or fail to grasp how to leverage the community and end up just annoying other contributors. In both cases our relationship with the community has taken a beating.

We've found that most students rise to the challenge presented in the open source development courses.

Properly supported, students thrive when presented with big challenges. Conversely, coddling students in terms of project scope or expectations almost certainly leads to failure.

By and large, the open source development courses have been successful for the majority of students. Notable project successes include:

1. APNG (http://animatedpng.com/): an extension of the PNG graphic format. While the PNG Development Group favored the use of MNG as the animated version of PNG, that standard proved difficult to implement effectively, and Mozilla wanted to try a lightweight, backward-compatible animated PNG format. Andrew Smith implemented APNG (http://zenit.senecac.on.ca/wiki/index.php/APNG) and his work has been incorporated into Firefox 3 and is also supported by Opera.

2. Buildbot integration: the Mozilla build system was adapted to work with the BuildBot automation system by Ben Hearsum (http://zenit.senecac.on.ca/wiki/index.php/Extending_the_Buildbot).

3. Plugin-Watcher: many Firefox performance problems are believed to originate with third-party binary plugins such as media players and document viewers. Fima Kachinski, originally working with Brandon Collins, implemented an API to monitor plugin performance, and created a corresponding extension to provide a visual display of plugin load (http://zenit.senecac.on.ca/wiki/index.php/Plugin-watcher).

4. DistCC on Windows: DistCC is a distributed C compilation tool originally written to work with GCC. Tom Aratyn and Cesar Oliveira added support for Microsoft's MSVC compiler, allowing multi-machine builds in a Windows environment (http://tinyurl.com/m3rj4b).

5. Automated localization build tool: there are many localizations that deviate in a minor way from another localization. Rueen Fiez, Vincent Lam, and Armen Zambrano developed a Python-based tool to apply a template to an existing localization to create the derivative version, which eliminates the need for extensive maintenance on the derivative (http://zenit.senecac.on.ca/wiki/index.php/Automated_localization_build_tool).

In addition, a number of graduates are now employed full-time by Mozilla and companies involved in open source as a result of their work. The open source courses have also led to a number of funded research projects in collaboration with open source projects and companies.

Lessons Learned

There are many lessons which students repeatedly take away from the open source development courses:

- it's important to persevere
- it's acceptable to share and to copy code, within the context of the applicable licenses, instead of guarding against plagiarizing or having your code stolen
- work in public instead of in secret
- tell the world about your mistakes instead of publicizing only your successes as there's a lot of value in knowing what does not work
- as a full community member you are a teacher as well as a student
- write down what you've done, and it will become a resource
- ask for help instead of figuring things out on your own
- key figures in open source are approachable, relationships are important and communication is critical
- code is alive

We've also learned that open source is not for everyone. The least successful students do not engage the community and attempt to work by themselves. However, even students who don't continue working with open source take an understanding of open source into their career, along with an understanding of how to work at scale which is applicable even in closed-source projects.

Finally, we've learned that open source communities and companies have a huge appetite for people who know how to work within the community.

Where We're Headed

The open source courses are growing and will continue to work within the Mozilla project. In addition, we began working with OpenOffice.org in fall 2008. Our Linux system administration graduate program (http://cs.senecac.on.ca/?page=LUX_Overview) is being revised to incorporate many of the principles that we've used in the other open source courses. LUX students will be working directly with Fedoraproject.org, but on a much larger scale as LUX projects will span three courses across two semesters. A build automation course was introduced into our system administration and networking programs in January 2009. This course will also be based on work within the Fedora project.

In order to effectively leverage our open source teaching, research projects, and partnerships, we've created the Seneca Centre for the Development of Open Technology (CDOT, http://cdot.senecac.on.ca) as an umbrella organization for this work.

Improving Student Involvement

Most open source communities actively welcome new contributors, but don't always make it easy to join. Steps a project can take to encourage contributors will improve student involvement:

1. Make it easy for new contributors to set up your build environment. Create an installable kit of build dependencies, generate a metapackage, or provide a single web page with links to all of the required pieces.

2. Create a central web page with links to basic information about your project that a new contributor will need, such as build instructions, communication systems, a list of module owners, a glossary or lexicon of community-specific technical terms and idioms, and diagrams of the software layers and components used in your products.

3. Create sheltered places or processes to enable new people to introduce themselves and get up to speed before being exposed to the full flaming blowtorch of the developer's lists and channels. This might include an e-mail list for new contributors, self-introductions, or an IRC channel for new developers.

In addition, in a course context:

1. Ensure that the community is aware of the course and course resources.

2. Feel free to join the student IRC channel, contribute to student projects as you would any other project, and read the student planet.

3. Contribute to learning materials on the course wiki.

4. Apart from recognizing the students as new community members, treat them as any other contributor.

Conclusion

Open source development is dramatically different from other types of software development, and it requires some radically different pedagogical approaches. A community-immersed, fully-open, project-oriented approach led by professor who is also a member of the open source community provides a solid foundation for long-term, sustainable student involvement in that community.

This article is based on a paper that was presented at Linux Symposium 2008 and published in the 2008 Proceedings, Volume 2 (http://www.linuxsymposium.org/2008/ols-2008-Proceedings-V2.pdf). A copy of the original paper, along with related resources, is available from the author's website (http://chris.tylers.info/ols2008).

Chris Tyler is a programmer and Linux network administrator with a focus on the X Window System and LAMP. He has programmed in two dozen different languages over the past 20 years, and now teaches at Seneca College, Toronto.

"sharing single user application over distance"

Definition of SHARE
http://grouplab.cpsc.ucalgary.ca/Projects/ProjectShare

Many work environments require collaborative writing and editing of documents in diverse formats. In simple cases, there is essentially one author who receives approvals and comments from others. In other cases, the document is genuinely collaboratively authored using asynchronous or synchronous methods. A common way to collaboratively edit a document is to exchange draft versions between authors via email. This method introduces the possibility of conflicting changes and missed contributions as well as a significant burden as all members of the team are responsible for version control. The principle difficulty is that independent changes can be made to different versions, which must later be reconciled manually. It is also difficult to determine when and why a change was made.

This article introduces TellTable (http://www.telltable.com/), an open source system designed to allow single-user software applications to be managed in a collaborative manner. We will discuss current collaboration models, the technical aspects of the TellTable software framework, security issues in its implementation, and tests of performance.

Collaboration Models

While collaborative writing and editing solutions have been offered since the 1970s, people prefer using single-user applications and distributing copies via email (http://tinyurl.com/m7dcck). Recently, designers have taken advantage of the Web's popularity in the hopes of supplanting email's popularity. We see three writing and editing collaboration models offered on the Web:

1. The upload/download model: a web server is used as a document repository. Group members upload and download documents to this repository, but edit on their personal computer, using a single-user application. One example of this approach is BSCWi (http://public.bscw.de), although that solution now offers online editing.

2. The web native model: a special editor is built from scratch to work through a web browser. The resulting document is often in HTML format. There are numerous examples of this approach, including wikis, SynchroEdit (http://synchroedit.com), WriteBoard (http://www.writeboard.com), NumSum (http://numsum.com), gOffice (http://goffice.com), and Google Docs and Spreadsheets (http://docs.google.com/).

3. The hybrid model: a normally single-user application such as OpenOffice is adapted to work through a web browser. Examples of this include TellTable, coWord (http://cooffice.ntu.edu.sg), coPowerPoint (http://cooffice.ntu.edu.sg/copowerpoint), and coStarOffice (http://www3.ntu.edu.sg/home/ashfshen/costar).

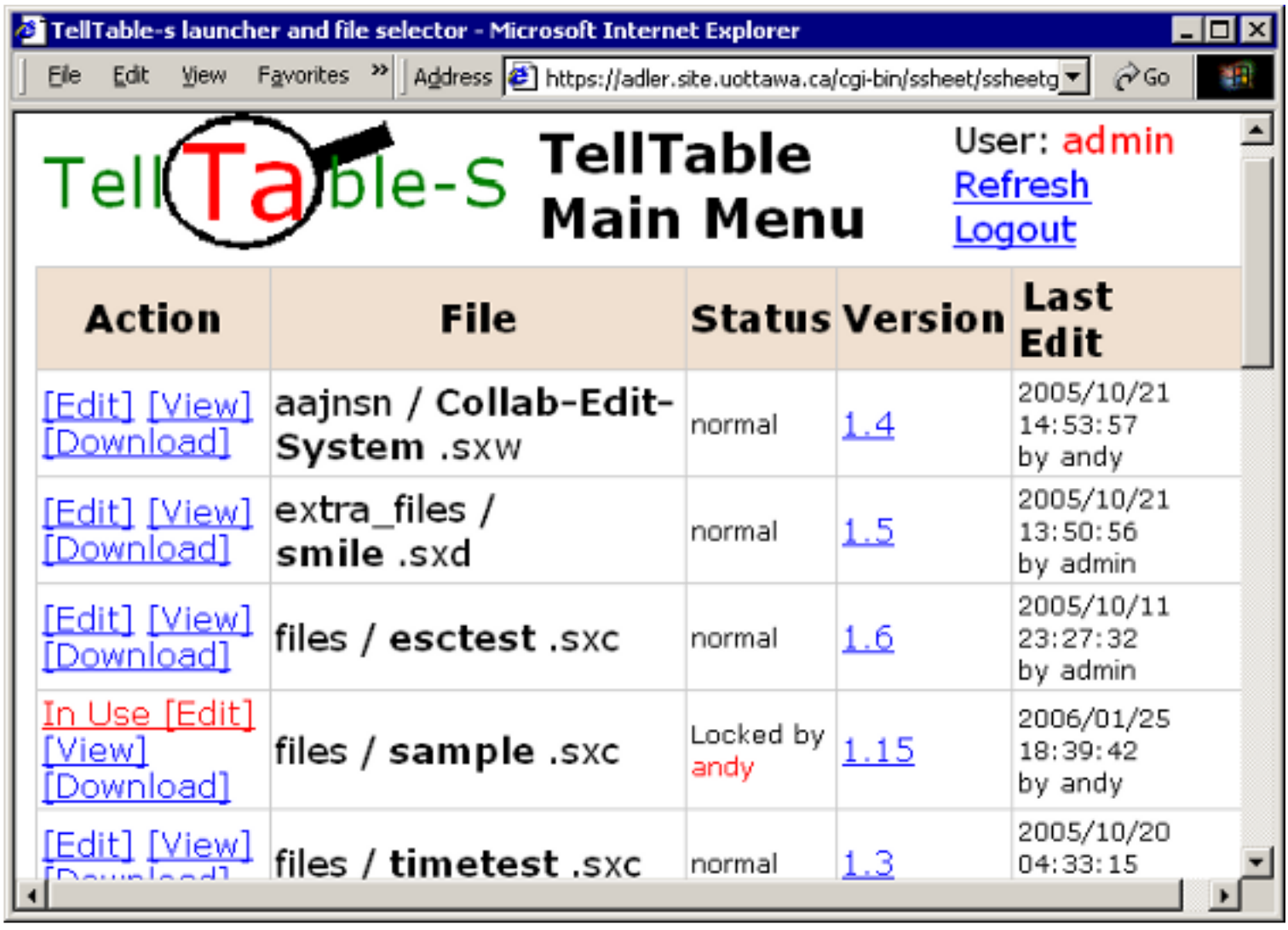
In the hybrid model, the use of a web browser to provide the display to the application allows users to employ a familiar interface to access document and application launching functions. It is feasible to use programs other than a web browser to display the data. Other non-web-based collaborative editing models include peer-to-peer and client-server models such as MoonEdit (http://moonedit.com/), SubEthaEdit (http://www.codingmonkeys.de/subethaedit/), and Groove Networks (http://www.groove.net).

TellTable

With TellTable, underlying single-user software, such as OpenOffice, runs on a server. Users log into the server with a browser and view and interact with that software via a browser plug-in. This approach provides an easy migration path to users familiar with single-user applications while providing the benefit of a server that wraps the applications in a collaboration environment. This is consistent with our philosophy that the principle challenge with the design of collaborative systems is ease of use. Participants in a hard-to-use collaborative system will tend to resort to emailing private documents between each other.

TellTable was initiated in 2002 as an approach for spreadsheet auditing. By running the spreadsheet software on a server, all user interactions could be captured and centrally managed. Subsequently, TellTable was expanded into a framework for a general collaborative system. The server component of the project was licensed under the GNU Lesser General Public License (http://www.opensource.org/licenses/lgpl-2.1.php) in March 2004, and is distributed from Sourceforge (http://telltable-s.sourceforge.net). We are currently pursuing various enhancements to its functionality and are actively interested in collaborating with others on its further development.

Figure 1: TellTable's File Access Screen



For the user, TellTable functions like a web application. The user enters the server URL into an Internet browser and is presented with a login page. After entering a username and password combination, the user is presented with a screen showing the current status of all files to which they have access. At the lowest level of privilege, a user will be shown the file name, latest version number, and date/time and user name of the last file edit. If another user has chosen to "Edit" a file, it will be marked as "Locked", unless the user has chosen to share the session in which case it will be marked as "Shared". User privilege levels may be set to allow other functions such as "Audit" or to allow access to the version history of files. Figure 1 shows a view of the file access screen.

The "Download" option will cause the browser to download the selected file to the local machine so it can be manipulated locally. Those changes occur outside of the TellTable framework and cannot be re-inserted into the file version history without administrative privilege. The "Edit" and "View" options use a Java VNC (http://en.wikipedia.org/wiki/Vnc) viewer applet to connect to a VNC server running the appropriate software. The server implements the chosen function, interacts with user input, and updates screen output.

By default, a user editing a file has exclusive write access to the file which corresponds to an asynchronous workflow. In order to allow synchronous editing, a user editing a file must click "Share File". This will allow other users, with permissions to edit or view the file, the ability to simultaneously access the document. Since the underlying application is single-user, all TellTable users will see the same view of the document and will have their keyboard and mouse interactions combined to the application.

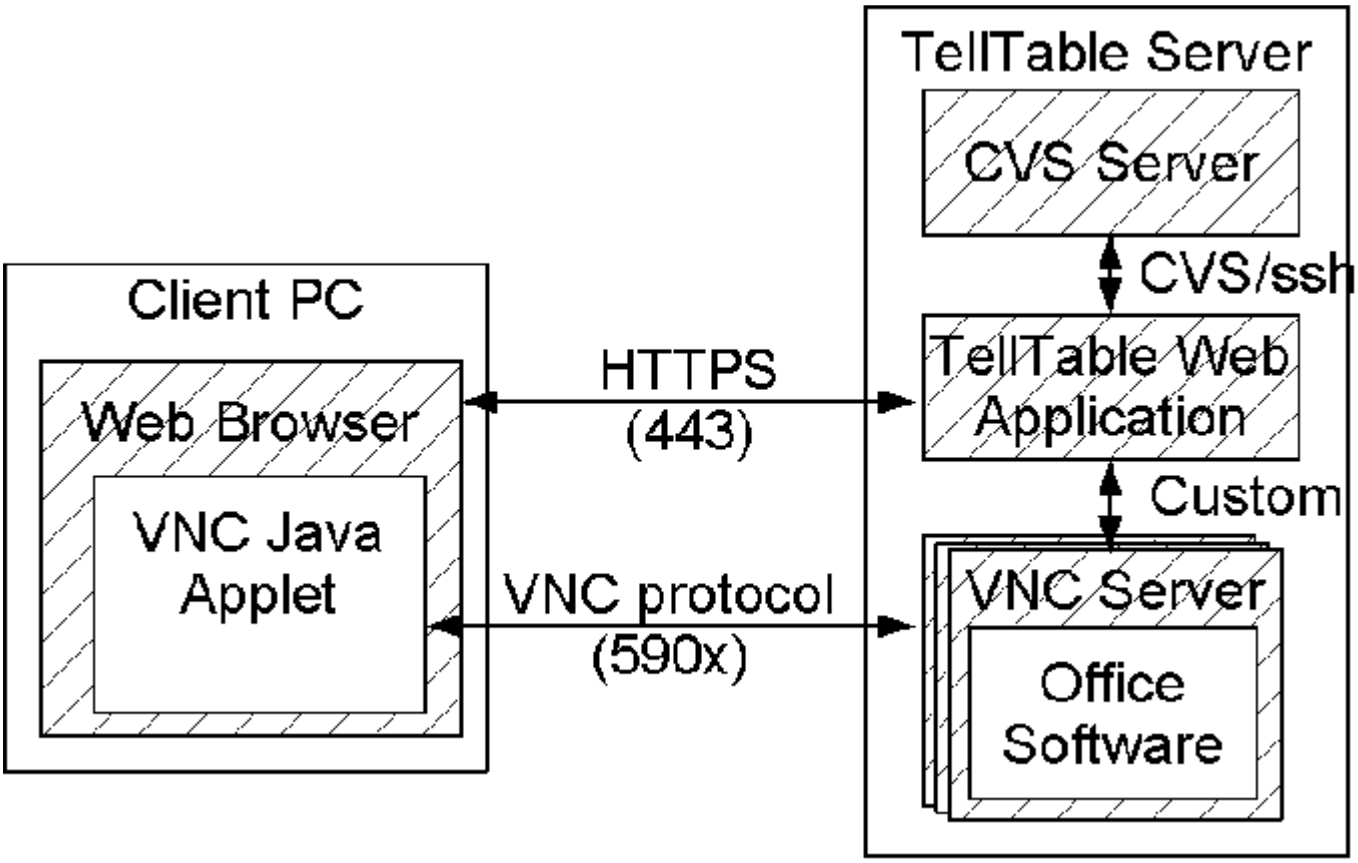
This means there is only one cursor and people must take turns to enter data. While synchronous editing with shared input may appear to be a source of conflict, our experience is that such editing is normally done in conjunction with a conference call between participants. The group tends to nominate one user to make the changes, while others watch and comment. Our experience is that when users compete for control of the keyboard and mouse, the user experience may become chaotic but the software remains stable.

Because of the constraints of the Java applet security model (http://java.sun.com/sfaq/), some operations function differently from their counterparts on a client workstation. First, the user needs to quit both the office software and the browser window. If the user closes only the browser window, the application is left running, and the user can access the application by logging back in. From a design standpoint, abandoned sessions need to be detected through a timeout and automatically closed. Second, copy-and-paste requires that we work around Java applet security that prevents applets from interacting directly with the clipboard of the client machine. Our design uses HTML input boxes to accept input from three sources of data: clipboard text, local files, and server files. In each case, the new data is copied to a read-only file on the server and opened as a sub-window of the OpenOffice document. This allows the user to select text from the uploaded document and paste it into the working document as required.

Open Source Components

A block diagram of the intercommunicating components of the server and a client computer is shown in Figure 2.

Figure 2: TellTable Design Components



TellTable is designed to offer cross platform support for Windows 98/2000/XP, Linux, and Mac OS X using Internet Explorer 5.0+, Mozilla 1.4+, Firefox, and Opera browsers. The client computer must have a graphical Internet browser with support for the HTTPS protocol and Java applets. The size of the VNC applet is configurable at install time with a default size of 900×550 pixels. This selection works well with screen resolutions of 1024×768 or higher, but can be a little inconvenient for lower screen resolution settings.

The core of the TellTable server is a pool of VNC server processes which may be distributed across several physical servers. At system boot time, a custom Perl program performs an initialization of the TellTable environment and initializes a TellTable server to accept and process commands.

The TellTable server runs under Apache on Linux. The web server components are primarily CGI scripts written in Perl. SSL encryption is used to protect login information. TellTable maintains a database of system activity using the BerkeleyDB format. Locking of the database between multiple CGI script invocations is implemented using the Perl module DB_File::Lock.

File versions are stored using the concurrent versions system (CVS) which allows branching, merging, and file differencing. TellTable's simple sequential progression of version numbers uses CVS to allow extraction of older versions and to maintain descriptive text logs with each version. Because TellTable manages conflicts using locking, CVS capabilities for branching and merging are not required. Although file differencing would be of great benefit to users, CVS was designed for plain text files while multimedia and other office software files are typically

binary. This means that the "diff" functions of CVS do not work as expected. A useful presentation of document differences would need to be determined at the application level, and some office software provide this function, but we are not working on interfacing to this capability at the moment.

Security Design

Collaborative work introduces security concerns that do not exist for the individual author. We categorize these as server vulnerabilities, client computer vulnerabilities, access control, and access level control.

Server vulnerabilities: collaborative systems require a networked server which, as a minimum, maintains a repository of file versions. Such network servers are potentially vulnerable to cracking through the applications running on the server. In more complex applications, the server is required to perform more complex operations, exposing it to a larger pool of possible attacks.

Client vulnerabilities: collaborative editing is not without risk to a client computer, even when exchanging draft versions via email. Most sophisticated office suites support macro languages, which are a popular vehicle for computer virus transmission. Since TellTable runs such applications on the server, it is important to protect it against viruses.

Access control: collaborative endeavours typically have a well defined group of participants. To ensure legitimate access, most systems require a login with a username and password. There are well known problems with password based systems which is a significant concern in a web-based system.

Access level control: group members often have different access privileges. Errors in setting access levels, especially in a complex system, can give users unwarranted additional privileges.

The TellTable server has a firewall configured such that only the HTTPS and VNC protocol ports can be accessed from the Internet and client machines. The firewall completely blocks outgoing network access from TellTable servers, meaning users and attackers will not have access to the server software running on the TellTable servers.

At server initialization, each TellTable process has a unique random code string embedded into it and stored in the Web server database. When issuing a command, the TellTable server calculates a SHA1 message authentication code based on the command text and the stored code. Commands with an invalid authentication code are ignored with an error. This code also serves as a one-time password for connections to the VNC server. This mechanism ensures that the user of a previous VNC session cannot eavesdrop on a future session since the authentication information will no longer be valid.

Perhaps the most vulnerable aspect of TellTable is the concern that users may be able to run arbitrary software as the userid used by the TellTable servers. This exposes the following vulnerabilities: i) users may snoop on other TellTable users; ii) users may attempt to hack the server; or iii) software may access the Internet to attack other machines or to act as a proxy or mail relay. In order to defend against such activities, TellTable has implemented three layers of defence. First, we attempt to prevent such access by careful configuration of the single-user software running on TellTable. For example, the options to browse the file system and run macros are disabled.

Second, to prevent arbitrary file system access, each process is run within a UNIX chroot which serves to isolate each process and the executable programs it is allowed to run from the rest of the TellTable file system. This helps prevent snooping of the TellTable server and other TellTable users. Third, the firewall is configured to block all outgoing Internet activity and only permit incoming activity on the VNC and HTTPS ports. This helps prevent use of TellTable for malicious Internet activity.

When the CVS repository is on the same computer as the Web server, file versions are stored under the userid of the Web server. If the CVS repository is on a separate machine, CVS commands are transported via SSH encryption. SSH authentication credentials are stored using the ssh-agent mechanism at Web server startup so the authentication information is not available to the Web server.

Performance

TellTable has been used by several workers at the University of Ottawa, the Communications Research Centre of Canada, the University of Vienna, and two independent teams linking Ottawa and Cardiff and Ottawa and Toronto. Applications include collaborative authoring of scientific presentations and articles, multimedia course material, and maintenance of course marks and documentation.

In the autumn of 2003, we performed our first pilot study of TellTable to manage spreadsheet files used to record course marks. Results showed that users appreciatived the features of the system, especially the ability to know one's changes would not be lost. Overall, usability was good. One concern was that responsiveness would suffer on slow Internet connections with high latency.

We were pleasantly surprised to find that even over slow links, TellTable was quite usable.

Some software bugs were triggered by patterns of usage of pilot users. For example, Hotmail opens links within a frameset that uses javascript to rewrite HTML to prevent breaking out of the Hotmail frame. Generally, the Hotmail rewriting would incorrectly rewrite the VNC applet frame, rendering it non-functional. The solution was to require logging into TellTable from a new browser window. This example also highlights the concern that such a technique could be used to capture passwords and other security information.

In terms of scalability, our tests indicate that the server memory requirements are relatively small compared to that required by the operating system and the spreadsheet data itself. Performance was calculated by performing simultaneous complex spreadsheet calculations. Results show that the TellTable server evenly distributes available computational resources with very little overhead. These results suggest that a moderately sized server with 1GB of memory should be able to support 10–20 TellTable sessions, depending on the requirements of the users. Since most uses of Office applications make sporadic use of computational power, it may be more efficient to use a powerful server for TellTable with less powerful client computers than individual powerful client machines.

Discussion

TellTable runs on a Linux server and supports clients using most popular operating systems and Internet browsers. We believe that the TellTable server is portable to other UNIX platforms, although we have no plans to do so. A port of the server to Microsoft Windows would require a significant rewriting, as Windows

does not easily allow multiple graphical sessions to run under different userids, as is required by TellTable. However, it is possible, using CodeWeavers Crossover Office (http://www.codeweavers.com), to run Microsoft Office software on Linux. In preliminary tests, we were able to run Microsoft Powerpoint remotely with Tell Table. However, it is unclear whether such use is permitted by the software license.

We have considered using a faster framework for dynamic web content than CGI, such as mod_perl (http://perl.apache.org). However, our current tests show that for reasonable loads of up to ten simultaneous users, the speed of the web server does not significantly degrade. Most delays in the web server are spent interacting with other system tools, such as the CVS or VNC servers. A possible annoyance with our design is the detection of real changes in files. For example, in Microsoft Word, opening a document, scrolling through and saving it, may result in a modified file. A version control system such as TellTable, will save these versions, unless software were written to detect such unchanged files. For the moment we have chosen to wait and see if this is problematic.

Initially, we considered dynamically creating a new VNC session when requested by the user. This approach proved infeasible because VNC servers require about 5 seconds to start, resulting in additional delay for the user. Worse, when the VNC server shuts down, its TCP/IP connection is left in the TIME_WAIT state and cannot be restarted for up to two minutes. An approach based on dynamically started VNC sessions would need to work around such timing considerations. Also, initiating VNC sessions for other userids requires elevated privilege for the web server, which may introduce security issues.

Another possible design approach considered was to maintain a VNC session for each system user, making security analysis easier. Unfortunately, this approach would require a large memory and processor capability to support a large number of users. Further, since each VNC server requires its own TCP port, it would require many open ports. As currently implemented, a VNC server is limited to 99 open sessions (TCP ports 5901-5999). Load balancing with multiple servers is another difficulty. If all logged on users happened to be allocated to the same VNC server computer, other machines would not be able to assist in supporting the computational load.

Our future technical work with TellTable involves improved usability and security enhancements, while simplifying some of the processes. One goal is to expand the set of applications we can launch and use with the infrastructure. We are also exploring ways to integrate workflow capabilities into the file-choice screen, since automation of the flow of files and information should enhance the utility of the infrastructure. As an open source project, TellTale benefits from the availability of ideas and code but requires input of time and resources from those choosing to join the project.

In conclusion, TellTable is a workable framework allowing single-user applications to be used collaboratively. This framework is open source and runs on inexpensive hardware. The TellTable approach benefits from considerable effort put into the development of user-friendly features in large software packages. Its value is in making it relatively easy to make such software function in a collaborative way. Pilot results show that users are generally able to use their familiarity with such software packages to work easily and effectively with TellTable.

This article is based upon TellTable: An Open Source Collaborative Editing System which is available for download from http://www.sce.carleton.ca/faculty/adler/publications/2006/adler-nash-noel-2006-collaborative-editing.pdf. The original article provides more in-depth coverage of TellTable's technical design.

Andy Adler is associate professor and Canada Research Chair in biomedical engineering at Carleton University in Ottawa, Canada. His research interests are in biometrics imaging and security systems, and development of non-invasive biomedical measurement technologies. Previously, he worked at several senior engineering positions. Andy Adler received a Ph.D. in Biomedical Engineering from the École Polytechnique de Montréal in 1995.

Dr. Sylvie Noël is a research scientist for the Communications Research Centre of Canada, where she works on the human factors of computer-supported cooperative work and collaborative virtual worlds. She has worked on projects on collaborative writing, video conferencing, social networks, and the incorporation of haptics and of emotions into virtual worlds. She is presently co-writing a book chapter on the difficulties associated with collaborative data analysis.

Dr. John C. Nash was Professor of Management in the Telfer School at the University of Ottawa until mid-2008. With a B.Sc. from the University of Calgary (in Chemistry) and a doctorate in Mathematics from Oxford, he has had a varied career in government, industry and academia. His books, articles and papers cover computation, statistics, forecasting, information science, risk management and quality and productivity improvement. He has also been a columnist for Interface Age, Scientific Computing Editor for Byte, and an editor or associate editor of several statistical journals. He remains active with several open source software projects, especially R and Gnumeric, and continues to offer his energy and expertise to others, both as a paid consultant, contractor or educator and as a volunteer to community projects, for example, as President of the Ottawa Canada Linux User Group.

Recommended Resources

Knowing What was Done: Uses of a Spreadsheet Log File
http://www.sie.bond.edu.au/articles/1.2/AdlerNash.pdf

Evaluating and Implementing a Collaborative Office Document System
http://www.sce.carleton.ca/faculty/adler/publications/2005/adler-nash-noel-2005-Collab-Office.pdf

Computer-Supported Collaborative Writing
http://hci.csc.kth.se/projectView.jsp?name=cscw

Obstacles and Solutions for Underrepresented Minorities in Technology

Copyright: Caroline Simard

From the Introduction:

Leading high-technology companies need employee diversity to remain globally competitive and innovative. Diversity leads to better group decisions, creativity, and innovation, as people from different backgrounds bring different skills and ideas to teams and companies. A diverse perspective creates enhanced market opportunities and better ideas. Women and men from underrepresented minority backgrounds are notably few in computer science and engineering disciplines. For women from underrepresented ethnic minority groups, the problem is even more serious.

http://anitaborg.org/files/obstacles-and-solutions-for-underrepresented-minorities-in-technology.pdf

Astronomical Software Wants To Be Free: A Manifesto

Copyright: Benjamin J. Weiner et al

From the Summary:

Astronomical software is now a fact of daily life for all hands-on members of the astronomy and astrophysics community. Purpose-built software to assist in and automate data reduction and modeling tasks becomes ever more critical as we handle larger amounts of data and simulations and doing steps "by hand" becomes less practical. However, the writing of astronomical software is unglamorous, the rewards are not always clear, and there are structural disincentives to releasing software publicly and to embedding it in the scientific literature, which can lead to significant duplication of effort and an incomplete scientific record...We advocate that: (1) the astronomical community consider software as an integral and fundable part of facility construction and science programs; (2) that software release be considered as integral to the open and reproducible scientific process as are publication and data release; (3) that we adopt technologies and repositories for releasing and collaboration on software that have worked for open-source software; (4) that we seek structural incentives to make the release of software and related publications easier for scientist-authors; (5) that we consider new ways of funding the development of grass-roots software; (6) and that we rethink our values to acknowledge that astronomical software development is not just a technical endeavor, but a fundamental part of our scientific practice.

http://mingus.as.arizona.edu/~bjw/weiner_software_CDH_FFP.pdf

June 5

Lead to Win Drives Innovation

Ottawa, ON

If you are serious about starting a profitable technology business in Canada's Capital region, we invite you to apply to the next session of the Lead to Win program. The program is free to qualified applicants. The objective is to create knowledge jobs, retain technology talent, and attract direct investment. If accepted, you must attend six-days of training scheduled for July 28 - 30 and August 25 - 27 of 2009. Each new business must be designed to grow so it can employ at least six knowledge workers over the next three years.

http://www.leadtowin.ca

June 8

Libraries: Creating the Future

Sault Ste. Marie, ON

The Arthur A. Wishart Library launched its Evergreen open source integrated library system with its Conifer Consortium partners: Laurentian University, the Northern Ontario School of Medicine and the University of Windsor. The University of Guelph plays a strong support role in the project, while McMaster University contributes to the ongoing development of the Evergreen system. The consortium has been working since July 2007 to build an enterprise class system that would meet the complex needs of academic libraries today and give libraries the freedom to develop the system for future need.

http://www.algomau.ca/view.php?page=news&parent=news&id=234

June 24

Annual Eclipse Release Now Available

Ottawa, ON

For the sixth year in a row, the Eclipse community has delivered its annual release train on its scheduled date. Galileo, the 2009 release train, is the largest ever release from the Eclipse community, comprising 33 projects and over 24 million lines of code. Over 380 committers from 44 different organizations participated to make this release possible. The new features in the Galileo release reflect three important trends in the Eclipse community: 1) Expanding adoption of Eclipse in the enterprise, 2) innovation of Eclipse modeling technology and 3) advancement of EclipseRT runtime technology. Each project has published "new and noteworthy" documentation for their specific release.

http://www.businesswire.com/portal/site/home/permalink/?ndmViewId=news_view&newsId=20090624005189&newsLang=en

Vancouver, BC

GeoWeb

July 27-31

GeoWeb conferences focus exclusively on geographic information systems, the Internet, and the economic potential associated with their convergence. GeoWeb 2009 will continue the tradition of focusing on the reciprocal impact of the Web and Geographic Information as well as the ever-increasing need for collaboration in light of global economic and environmental concerns. Representatives from both public and private organizations are invited to meet, discuss and learn about today's most innovative geospatial technologies.

http://geowebconference.org/

August 12-14

USENIX Security Symposium

Montreal, QC

Join researchers, practitioners, system administrators, and system programmers for the latest advances in the security of computer systems and networks.

http://www.usenix.org/events/sec09/

August 12-14

OpenEd

Vancouver, BC

A conference that focuses on open content and open educational resources.

http://www.openedconference.org/

Aug 25-27

World Congress on Privacy, Security, Trust and the Management of e-Business

Saint John, NB

This event will feature seven theme areas: privacy, security, trust, eHealth, HCI, eInnovation, and eGovernment.

http://www.unb.ca/pstnet/congress2009

Thinking about starting a new tech business? Then, think about

Lead to Win

"Lead to Win is for talented individuals who are serious about launching a new technology business. The program is free to qualified applicants – the next session begins July 28, 2009."

Professor Tony Bailetti, Carleton University

Lead to Win Alumni

David Vicary, Founder, Nakina Systems

"… no nonsense approach to the fundamentals of business …"

" … anyone launching a startup would find Lead to Win invaluable …"

Chuck Colford, President/Founder, Congruance IT

"… fundamentally relevant to launching, financing and operating any business …"

Jerry Everett, President/Founder, onconference

Lead to Win, part of Carleton University's Talent First Network, is focused on driving massive entrepreneurial activity

The Talent First Network is sponsored by Carleton University and the Province of Ontario.

Lead to Win is also supported by the City of Ottawa, the Ottawa Chamber of Commerce, OCRI, Developpment economique-CLD Gatineau , Arrow Electronics, onconference, and NRC-IRAP.

Apply now at www.leadtowin.ca

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 2009:	Tech Entrepreneurship Guest Editor: David Hudson
September 2009:	Business Intelligence Guest Editor: Mike Andrews, SQLPower
October 2009:	Arts & Media Guest Editor: Anthony Whitehead
November 2009:	Co-Creation Guest Editor: Stoyan Tanev
December 2009:	Bootstrapping Startups Guest Editor: John Callahan

Formatting Guidelines:

All contributions are to be submitted in .txt or .rtf format.

Indicate if your submission has been previously published elsewhere.

Do not send articles shorter than 1500 words or longer than 3000 words.

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.

Include a 2-3 paragraph abstract that provides the key messages you will be presenting in the article.

Any quotations or references within the article text need attribution. The URL to an online reference is preferred; where no online reference exists, include the name of the person and the full title of the article or book containing the referenced text. If the reference is from a personal communication, ensure that you have permission to use the quote and include a comment to that effect.

Provide a 2-3 paragraph conclusion that summarizes the article's main points and leaves the reader with the most important messages.

If this is your first article, include a 75-150 word biography.

If there are any additional texts that would be of interest to readers, include their full title and location URL.

Include 5 keywords for the article's metadata to assist search engines in finding your article.

Copyright:

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.