OSBR.ca

The Open Source Business Resource

MAY 2011

MAY 2011

PUBLISHER

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

EDITOR

Chris McPhee
chris.mcphee@osbr.ca

ISSN

1913-6102

Editorial

Chris McPhee

The editorial theme for this issue of the OSBR is Technology Entrepreneurship. We have invited entrepreneurs associated with the Technology Innovation Management program (TIM; http://carleton.ca/tim) at Carleton University to share their lessons and insights about growing a technology company during its early stages. The authors represent a range of entrepreneurial experience, from serial entrepreneurs reflecting on battles won and lost, to first-time entrepreneurs describing the early twists and turns of transforming ideas into ventures.

What is common to all the articles is the approach to entrepreneurship that is nurtured in the TIM program. The TIM program is a graduate program that distinguishes itself by offering three important benefits to its students: i) a Master's degree by research; ii) opportunities for personal brand development; and iii) practical, real-world experience. In particular, personal brand development and real-world experience are gained by applying the program's lessons and the products of the student's own research to assist early-stage technology companies. If the student is an entrepreneur, they have the added benefit of applying their research and learning to further their own opportunity. In this issue, entrepreneurs from the TIM program share some of the key lessons they have learned.

Michael Ayukawa, founder of Cornerportal, reflects upon the shifts in his entrepreneurial world view framework that came about from his participation in the Lead to Win business ecosystem and his graduate studies in the TIM program. By describing the transformation and adaptation of Cornerportal's strategy in response to the shifting world view of the entrepreneurs behind it, he highlights the transformative effect of collectives on an entrepreneur's view of their environment and options.

Natasha D'Souza, founder of Virtual EyeSee, describes a problem for which she is currently developing a business opportunity to solve: helping parents provide adequate and appropriate support to children with attention deficit hyperactivity disorder (ADHD) or autism. She uses this problem as a case study to illustrate how the implementation of a facial emotion recognition software application might be substantially different depending on the development and commercialization approach used. The article focuses on the impact of leveraging collectives to develop compelling solutions that increase the likelihood of entrepreneurial success.

Fred Dixon, CEO of Blindside Networks, share key lessons learned while balancing the demands of building an open source business and nurturing the open source project that the business depends on. He shares lessons from BigBlueButton, an open source web conferencing system developed in the TIM program, and he shares lessons from Blindside Networks, a company that was spun out of Carleton University to provide commercial support to academic institutions.

Arthur Low, founder and CTO of Crack Semiconductor, retraces the history of key advances in the integrated circuits and electronic design automation tool industry to show that a shift from proprietary to open source tools now means that viable business models exist for small companies to create advanced silicon intellectual property. He provides two case studies

to show how the shift to open source has made this high-end technology accessible to low-budget startups.

Igor Sales and Aparna Shanker outline the business opportunity they are developing within the TIM program to help bring together freelance Android developers and the software development firms in need of them. The key aspects of their opportunity are: i) the creation of a strong collective of Android developers and companies and ii) the means to prove the expertise and reputation of developers within the collective.

We encourage readers to share articles of interest with their colleagues and to provide their comments either online or directly to the authors.

For the upcoming June issue, we continue the theme of Technology Entrepreneurship. We encourage any entrepreneurs who wish to share their insights and lessons to submit articles for this issue before May 15th. In July, we focus on Women Entrepreneurs and welcome submissions that shed light on the particular challenges of increasing the number of women in founding and leadership positions. Please contact me (chris.mcphee@osbr.ca) if you are interested in submitting an article for either of these themes; we also welcome general submissions on the topic of open source business or the growth of early-stage technology companies.

Chris McPhee

Editor-in-Chief

Chris McPhee is in the Technology Innovation Management program at Carleton University in Ottawa. Chris received his BScH and MSc degrees in Biology from Queen's University in Kingston, following which he worked in a variety of management, design, and content development roles on science education software projects in Canada and Scotland.

Shifting an Entrepreneur's World View

Michael Ayukawa

"When one door closes, another opens; but we often look so long and so regretfully upon the closed door that we do not see the one which has opened for us."

Alexander Graham Bell

As an entrepreneur, you continually test your decisions by gaining feedback: from your customers and your investors (or lack thereof). This process of ongoing feedback is how an entrepreneur learns to shape their opportunity to accommodate their new knowledge of the environment. But this activity is very dependent on the "world view framework" of the entrepreneur. What may seem to be important turns out to be noise and important signals are dismissed. This article describes the special value for an entrepreneur of frameworks grounded in theory in general and the value of the framework of business ecosystems from two perspectives: as a member of a business ecosystem and as a creator of a business ecosystem. These two perspectives fundamentally affected the direction of adaption for our product and reshaped how we approached our (ad)venture.

Introduction

A story allows the author to rationalize decisions based on hindsight. For a story told by an entrepreneur, this might be a fair tradeoff given that they make many of their decisions in an environment of high uncertainty (McMullen, 2006; http://tinyurl.com/44pjgd9). This is my story.

Freshly unemployed and lacking the foresight to take a pre-emptive vacation, my first partner and I rapidly churned through many mashups of old and emerging technologies connected to the nascent wave of connecting data and people to location and objects. Without any hot prospects for early and meaningful revenue, my partner departed to enjoy the practical reality of a salaried position. Continuing to move forward with a new partner, what soon emerged was a product that we coined "Social Signage" (http://tinyurl.com/4x2om33). I would soon learn that our process was as a classic case of technologists working hard to find the right problem to fit a solution.

The Before Picture

Despite our best efforts, it was not a pretty picture. Through volunteer labour, applied research students from Algonquin College (http://algonquincollege.com/appliedresearch/), the Ontario Self-Employment Benefit Program (OSEB; http://milkshake.ca/oseb/oseb_eng.html), and dogged perseverance, we had finally achieved a major milestone: the installation of our first digital screen in a café (http://www.umicafe.org/). The euphoria of this moment quickly passed and we faced the cold reality shared by many of our fellow entrepreneurs starting up in a downturn: we would starve before this would turn into anything meaningful. Our barrier was that our customer value scaled with the size of our net-

work. So, as a small network, without the capital to subsidize early adopters, it was going to be very tough. This I would subsequently learn was entirely predictable from Metcalfe's law (http://tinyurl.com/6czzed).

Despite the best intentions and support from many programs, we were stuck. We were not lost, we knew exactly where we stood, but it was obvious that our chances of survival were slim, given the barriers we faced to move forward in the metaphorical jungle.

Opportunity and Serendipity

Quite fortunately, but not entirely fortuitously, there was a timely confluence of learning and practical necessity. In parallel to this seemingly Brownian motion of creative activity, a continuous thread of more formal entrepreneurship education was already well underway. The initiation of this thread had already taken place months before with acceptance into the freshly re-minted Lead to Win program (http://leadtowin.ca/). Like all roads leading to Rome, several months later I found myself enrolled at Carleton University in the Technology Innovation Management (TIM; http://carleton.ca/tim/) Master's program.

Attracted to the TIM program for highly pragmatic reasons, I was soon introduced to, and became totally absorbed in, the theories of technology management, business ecosystems, and multi-sided platforms. Elements of this journey can be seen in my archive of OSBR submissions (http://tinyurl.com/6dkgelx). There were two very practical aspects of this learning: one was the concept of business ecosystems as described by Dr. Tony Bailetti (2010; http://tinyurl.com/32jlwm7) in the OSBR. But the other was the value to an entrepreneur of conceptual frameworks based on academic theory, rather than popular literature including management fads and entrepreneurial best practices.

As Carlile and Christensen (2005; http://tinyurl.com/5vb6b2) explain: "The external validity of a theory is the extent to which a relationship that was observed between phenomena and outcomes in one context can be trusted to apply in different contexts as well." Therefore, a conceptual framework based on theory was more powerful to us than our working hypothesis that was, for all practical purposes, derived from anecdotal evidence. Viewing our situation through a lens grounded in theory helped us clearly see the nature of the barriers we faced. We were like the blind men from a well-known parable finally agreeing that what we were dealing with was actually an elephant (http://wikipedia.org/wiki/Blind_men_and_an_elephant). Once we knew it was an elephant, we could make better decisions.

What I did not anticipate was the effect of internalizing the framework of business ecosystems and the principles of multi-sided platforms. This enabled a true paradigm shift. By looking through this lens, the world suddenly becomes very different to an entrepreneur. A new path to reach critical mass for our venture emerged. Being able to reframe the product opportunity in terms of an environment structured as a multi-sided platform allowed us to abandon our traditional standalone, push business model and embrace a pull model where the the barriers of money and resources no longer stood directly in our path. This was very exciting to discover and gave us new confidence and hope.

There is of course another aspect to business ecosystems, or collectives, and this relates to the value to an entrepreneur of belonging to such a collective. Being a member of a collective such as Lead to Win or the Carleton Entrepreneurs (see Bailetti's article in the April issue of the OSBR; http://tinyurl.com/3tjjmyt) simply lowers the risk and makes it less expensive (or more profitable) for an entrepreneur to compete or collaborate with the incumbent. It does this by making it easier to access expertise, increasing

the stakes for cooperation, and leveraging the stronger brand of the collective. This concept of operating inside or outside a business collective is an interesting extension of the classic framework of Gans and Stern (2002; http://tinyurl.com/3g285bd), where they describe a framework that links business strategy for a technology entrepreneur to that of the commercialization environment for their opportunity.

There are several points to take away from this experience:

1. A mental model of the business environment not only frames your opportunity and therefore the design of your business model, but it changes how you observe and discriminate the signals that come from the market.

2. A model based on theory is likely more robust and plausibly extensible to your context, which as a startup will by definition have some element of novelty.

3. Framing the environment through the lens of a multi-sided platform gives the entrepreneur new possibilities for their venture to break free of the constraints of a standalone push model. This is huge for a bootstrapping entrepreneur.

4. Reframing from a traditional, standalone push model to a multi-sided platform, pull model can be very challenging for individuals who have historically found success in the traditional model and have not yet internalized the principles of a business ecosystem.

5. It is lower risk and less expensive for an entrepreneur to start a venture as a member of a business ecosystem. The entrepreneur can take advantage of the resources and diverse perspectives of members inside the ecosystem. More and different opportunities emerge for growth and revenue generation. As a collective, the brand of the ecosystem is generally more powerful and valuable than that of a barely nascent firm.

The After Picture

For our company, Cornerportal (http://cornerportal.com), this was transformative on two levels. We had found confidence based on an understanding of our position in the marketplace and a possible path for our success. Success was by no means guaranteed, but we had a better sense of what direction we could take and make progress with the limited resources at hand. This, in a sense, kept us in the game. This, in turn, gave us the time to rediscover the environment through the new lens of multi-sided platforms and complete our transformation to a pull model of business development.

But even with a transformation of our business model, things were still not right. Although we were able to define a better business model for our opportunity, it was not powerful or compelling enough to gain any real momentum. Or put another way, we still struggled to justify continuing to invest our personal resources into the project. It was time for a serious review.

We challenged ourselves to look at the environment with new eyes and to question all previously held beliefs. This allowed us to jettison what we thought was the cornerstone of our product and value proposition: the on-premises digital screen. We discovered it was an artifact of our journey and where we thought our core value was rooted. Through this process, we uncovered a previously missed opportunity and what has now become our primary focus. It was particularly upsetting was that we had earlier in our journey dismissed key elements of the approach that we were now embracing.

But again and in retrospect, this was perhaps not entirely surprising. The emergence of the new approach was again connected to parallel learning in coordination of deal processes for business ecosystems (see Ayukawa, 2011; http://tinyurl.com/6gfy9hq). Fundamental to this work was the notion of coordination through shared objects (Bailetti et al., 1994;

http://tinyurl.com/6655ulh). Adding this model to our thinking helped not just identify the new opportunity but also to bring clarity to the potential scope of this opportunity. This in turn allowed us to obtain early-stage funding in a remarkably efficient process.

There are a few points to make from this experience:

1. A change in mental model or framework of the environment can make what seems old, new again. Forcing a change in your mental model may be the first step to find a new direction.

2. Take this as an opportunity to challenge closely held beliefs and what you believe to be core to your business. You may be surprised at what you discover.

3. Stay in the game if you can. Never give up. Opportunities come to those still in the game.

Our New "Stikky" Product

We expect to release our new product, "Stikky", in May 2011 via the Apple Appstore. Stikky will make it easy for any ad hoc group of individuals who, through a specific shared context, find it valuable to contribute content or sentiments. Or put another way, we make it easy and productive for people who might not otherwise connect, to share their perspectives about something they care about. All from their mobile device.

The difference is the practical focus or filter that comes with the context of the shared object, be it a common objective, an experience, or a passion. The value of this service now becomes intrinsically linked to the value of the shared object from either a practical or emotional perspective. One additional outcome from such a highly contextual timeline of contributions is a rich storyline that would otherwise be difficult to assemble after the fact.

In a powerful way, this also solves the scaling problem of how to productively extend the reach of your functional personal and professional network without the noise associated with a large, but unstructured network. In fact, people do not have to share personal contact information, since the connection is made through the shared object. There is no need to “be friends” or be “LinkedIn” to make this happen.

What makes this particularly exciting is the potential diversity of applications. For example:

1. Imagine being able to capture your child’s sentiments over time and easily attach them to their favourite stuffed animal or toy. No syncing, no file folders, no shared drives.

2. Imagine your ad hoc group who came together to help reduce vandalism in the neighbourhood. Observations and snapshots can be easily captured and shared, together with location, to create a visual map of activity. No logging in, no email lists, no wasted time.

3. Imagine being able to easily visualize the connections (or lack thereof) between the players that have played a role in a fragmented but interdependent project. No special project software, no data merging, no waiting for reports.

Conclusion

As entrepreneurs, we have to hope that we will make better choices and learn more rapidly than our competitors. Entrepreneurs who are members of a business ecosystem have a strategic advantage due to their network of trusted relationships and affiliation to the whole. It simply has lower risk, is less expensive, and has a higher probability of success. For Cornerportal, this is the business ecosystem of Lead to Win.

The framework of a business collective can be transformative to an entrepreneur's view of the

environment and the options they have to overcome the traditional barriers presented by a push model of commercialization. Looking at the environment through such a lens can be difficult for those who have not internalized the theory and have historically found success in the traditional approach. For Cornerportal, this was the result of formal course studies in the TIM program at Carleton University.

Startup companies that have access to, and a meaningful engagement with, the resources from academic institutions have an advantage based on exposure to theoretically supported frameworks that can help them discern signal from the noise and thus better accommodate the high degree of uncertainty that is part of being an entrepreneur. For Cornerportal, these are the academic resources of the TIM program.

Michael Ayukawa is a Master's student in the Technology Innovation Management program at Carleton University and plays an active in several emerging business ecosystem projects, including co-founding Cornerportal Inc., a company that is committed to bring economic opportunity to more individuals in more communities worldwide.

Leveraging Collectives as a Technology Startup

Natasha D'Souza

"Without deviation from the norm, progress is not possible."
Frank Zappa

Entrepreneurs face a daunting challenge in turning a solution to a compelling problem into a viable business. Recent research into multi-sided platforms and collective action has highlighted an approach that may enable entrepreneurs to lower the risk of a new venture and increase revenue by delivering value to all stakeholders in a collective, not just to the company itself. However, the shift in thinking required to apply this new approach is a challenge of its own.

In this article, we provide an overview of both the new and traditional approaches to development and commercialization. Next, we describe a problem for which we are currently developing a business opportunity to solve: helping parents provide adequate and appropriate support to children with attention deficit hyperactivity disorder (ADHD) or autism. We then use this problem as a case study to illustrate how the implementation of a facial emotion recognition software application might be substantially different depending on the development and commercialization approach used. Finally, we describe the key lessons learned and next steps in developing this business opportunity.

Introduction

In the September 2010 issue of the OSBR, Tony Bailetti described a new model of development and commercialization (http://tinyurl.com/3287e9q), which he labelled Model C. The Model C approach is in stark contrast to the traditional standalone approach (or Model A), which "pushes a supplier's products, services, and solutions to customers, either directly or through intermediaries." The Model C approach encourages companies to interact with multiple stakeholders to rapidly co-create products and services. The focus of Model C is to, "create new things that deliver value to customers and to all the organizations that contribute to the company's development and commercialization initiatives." This approach can be very effective, particularly if used to harness the power of collective action. Collectives are groups of people and organizations brought together to achieve a common goal; they "harness diversity to produce significant system-level outcomes" (Bailetti, 2011; http://tinyurl.com/3kngqbn).

Bailetti's key criticism of the traditional Model A approach is that the required development and commercialization process is too time consuming and expensive to lead to long-term success in today's environment. While he argues that the Model C approach is an entrepreneur's best bet for generating revenue in the short and long term, this new approach requires a new way of thinking. In this article, we use a case study of the author's business opportunity to illustrate how the Model A and Model C approaches lead to different implementations of the same solution. In the following section, we describe the problem to be solved.

Case Study: ADHD and Autism

The inability to interpret facial and tonal emotion lies at the core of several disorders affecting children, including ADHD and autism (http://tinyurl.com/5wb6bg9). Children affected by this problem – known as facial affect recognition disorder – face a number of significant challenges that may be unrelated to their intellect, but affect their ability to interact effectively with their peers. These children thus react differently to social situations. They may be aggressive, argumentative, susceptible to meltdowns, and generally challenging to interact with. These behaviours may lead them to be ignored by their peers or bullied. In the case of ADHD, research has shown that these children have higher rates of substance abuse in their teen years compared to other children (http://tinyurl.com/6369net).

ADHD affects 3 – 5% of children (http://tinyurl.com/5srhfq4), while autism and related disorders affect 0.1 – 0.2% of all individuals (http://tinyurl.com/5rowlhs). In addition to these numbers, we must also consider the impacts on the parents and caregivers of children affected by these disorders.

When a diagnosis of ADHD or autism is reached, parents seek solutions to help their child. They are surprised to find that the existing treatment options are heavily focused on medications, with behavioural therapy playing an important, but poorly supported, role. Parents also report that the health care system is severely backed up, with long wait times for clinics and other treatment services. When spots become available, regular visits put added pressure on the time and monetary budgets of families, particularly if they live in rural locations. Despite best efforts, schools are unable to offer much beyond independent learning plans. Together, these factors leave parents in a difficult situation; they are desperate to help their children, but face a stressful challenge and struggle to provide adequate and appropriate support.

Parents often turn to a limited selection of tools designed to help children develop facial emotion recognition and social skills. These tools include posters, flash cards, books, animated computer games, puppets, and role-playing scenarios. However, evaluating the child's progress with these tools is difficult and subjective; parents often cannot identify which specific areas are showing improvement and which ones need to greater attention.

Proposed Solution and Competing Implementation Approaches

To solve the problem outlined above, we propose the development of a software application that will deliver facial emotion recognition training using tablet computers.

The most obvious business model for this solution would be to develop the application and then sell it to parents directly or through an app store. Ideally, one or more healthcare providers would be hired as consultants to provide advice on the features, including perhaps reporting functions for parents to track their child's progress. This implementation would be a typical outcome of the traditional standalone approach (Model A). However, for the solution described here, the Model A approach minimizes the value of the application because it only addresses the facial emotion recognition aspects of the problem. As a small company operating under this approach, any attempt to try to do more than build and sell the application might be interpreted as taking the focus away from the core of the business. Selling applications is a viable business strategy, but it is likely difficult to sustain success in the long run.

In contrast, following the Model C approach to this problem results in a very different implementation and business model. When viewing this solution through a model C lens, we must consider that this solution adds value to the child, parent, healthcare provider, and techno-

logy providers. Acting as a collective, these groups can co-create a significantly more valuable solution to improve the lives of children with ADHD and autism. Considering the interests of other participant's products and services creates opportunities to develop an integrated solution. For example, we can interact with parents, healthcare providers, and technology providers to add data-capture capabilities that will track the child's progress, analyze the data, and send reports to a designated healthcare provider. In turn, the healthcare provider could use the application to view reports and discuss the results with the parents using integrated teleconferencing features.

The Model C solution goes beyond just adding value to the parents in the form of time saved and additional information about their child's progress; it also adds value to the healthcare providers, who save time and gain a new support service and it adds value to the solutions technology providers can offer to healthcare staff and consumers. For the technology startup, this equates to earlier commercialization, lower initial costs, and a greater number of revenue opportunities. Below, we further describe the value propositions to each member of the collective, including also researchers and change agents:

1. Children: For the child, the solution provides a fun and exciting game that they look forward to playing by themselves. They find they are allowed more time on the computer, without a lot of negotiations with their parents. Their social life improves along with their relationships with their parents, even if they are unaware of the cause.

2. Parents: Parents finally get a solution that their child can work on independently. It is convenient and affordable. Because of the teleconferencing service, they no longer need to drive all over the city for meetings with healthcare providers. They can monitor their child's progress and any improvements they make. Their stress levels decrease because they now have an effective solution that is based on their needs.

3. Healthcare providers: Healthcare providers now have the ability to offer new services that complement their existing ones. They save time because their efforts spent developing the application now scale to the benefit of other users. They have a way to measure the progress of the child through the automatic data capture, analysis, and reporting capabilities. In addition to this, they no longer have to spend their time generating progress reports. They can now focus their time on working with the child to improve those aspects of social skills that are deficient. They also benefit from positive interactions with parents who are now less stressed about their child's condition.

4. Technology providers: This solution leverages several existing technologies and open source platforms. Providers will be able to increase their revenues and exposure in new market segments.

5. Researchers: With input from researchers into the development of the application, it can become a means to gather data and test theories of facial emotion recognition. Incorporating their findings and feedback into the product will be of significant value to the future users of the application.

6. Change agents: Opinion leaders act as agents of change. These individuals are key influencers who help change social norms and accelerate change (Valente and Pumpuang, 2007; http://tinyurl.com/6h5ygzr). In this case, opinion leaders influence new service offerings and technology implementations. As members of the collective, they are able to perform their role more effectively. Providing them with the necessary education and training and converting them to champions will go a long way to increasing the adoption of this product. This is significantly more effective than client testimonials.

Lessons Learned and Next Steps

Identifying the problem described in this case study was an iterative process that started with a hunch, and then involved a close examination of all possible stakeholders and the key issues that affected each of them. The challenge was to figure out if this was a problem for one person or for many. Examination of the research relating to different aspects of this problem confirmed that this was indeed a solution for many, globally.

Having an understanding of the pain points for the various stakeholders was crucial to the development of an effective theoretical solution. There were many solutions possible, but being able to tie the solution together to address the pain points of many stakeholders and change the whole system as a whole made for a compelling solution.

As a business opportunity, this solution is in its very early stages. The next step is to take this idea and quickly validate it with minimal cost. A prototype is scheduled to be built in collaboration with representatives of key stakeholder groups. The decision of whether or not to proceed with this opportunity will be based on feedback on the prototype.

Conclusion

Children that have ADHD or autism no longer have to be bullied or isolated by their peers. This solution addresses the core of the problem and gives them the tools necessary to socialize effectively with their peers. It gives them an activity to do independently while reducing the stress of their parents by eliminating the need to take time off work to travel. Healthcare providers and other stakeholders also benefit by offering new and more effective solutions. By applying the Model C approach to this problem, a more compelling solution is achieved.

Natasha D'Souza, founder of Virtual EyeSee, has over 15 years of hi-tech experience working for Fortune 500, mid-sized, and startup companies. She has a degree in Electrical Engineering and is currently a graduate student in the Technology Innovation Management program at Carleton University. She is a regular guest speaker who is passionate about technology and inspiring people to develop unique solutions to complex problems.

Lessons from an Open Source Business

Fred Dixon

"It's a brilliant surface in that sunlight. The horizon seems quite close to you because the curvature is so much more pronounced than here on earth. It's an interesting place to be. I recommend it."

Neil Armstrong

Creating a successful company is difficult; but creating a successful company, a successful open source project, and a successful ecosystem all at the same time is much more difficult. This article takes a retrospective look at some of the lessons we have learned in building BigBlueButton, an open source web conferencing system for distance education, and in building Blindside Networks, a company following the traditional business model of providing support and services to paying customers. Our main message is that the focus must be on creating a successful open source project first, for without it, no company in the ecosystem can flourish.

Introduction

The common portrait of the entrepreneur is someone dreaming of the future, a future somewhere over the horizon. The entrepreneur is portrayed as someone who is constantly looking for that "next big thing" and who is willing to work longer, harder, and with more focus than others to realize it. However, for an entrepreneur to be successful, at some point the dreaming must shift to focus on the near-term horizon of necessary activities to build a viable business. Successful entrepreneurs live in the present, where the horizon is very close, and the sunlight cast from the company's cash flow is very, very bright.

Shifting to a focus on the near-term and maintaining focus is a difficult and complex challenge. At Blindside Networks (http://blindsidenetworks.com), we have faced this complexity many times. In this article, we share some of the key lessons we have learned over the past three years.

Blindside Networks and BigBlueButton

Blindside Networks was spun out of Carleton University's Technology Innovation Management (TIM; http://carleton.ca/tim) program to provide commercial support to BigBlueButton (http://bigbluebutton.org). BigBlueButton is an open source web conferencing system for distance education. Initially, it was developed by the students and faculty in the TIM program. Richard Alam, a co-founder of Blindside Networks, was the first student in the TIM program to complete a thesis on how to make money from open source. He started the project (http://tinyurl.com/3zu4f2z) and continues to be one of BigBlueButton's lead developers. The idea for BigBlueButton was simple: reduce costs by providing a viable open source solution for giving remote students a high-quality learning experience. The challenge was to create a viable business around that solution.

At Blindside Networks, we follow the traditional open source business model: make the product

freely available and charge for support and services. More specifically, start with a business model that is based on professional services and, over time, gradually shift a portion of our development efforts to offer complementary products and services that are proprietary. Along the way, we partner with other companies to offer a whole product to customers.

As an open source business, Blindside Networks must balance its entrepreneurial activities with its community activities. Unless BigBlueButton itself actually solves problems around distance education (i.e., unless it "works"), the opportunities for Blindside Networks to grow its revenues from services and support are limited. Furthermore, we must nurture the project and encourage participation from many users, developers, and customers from diverse backgrounds and interests. This creates a very healthy ecosystem that, in turn, creates entrepreneurial opportunities for Blindside Networks and others.

As entrepreneurs, the challenges we face at Blindside Networks are threefold:

1. Leading an open source project that solves a real-world problem.

2. Creating an ecosystem for the project that attracts others to improve it.

3. Building a viable business providing support and services to the ecosystem.

In the following sections, we share our experiences in the hopes that others can benefit from knowing what has worked well for us and what we would have done differently if we could start over. Of course, we cannot start over, but we hope that reflecting on these lessons will help others face similar challenges. We include lessons from both the perspective of a business and the perspective of an open source project.

Lessons Learned Operating a Business

Lesson 1: Focus on one market segment. At Blindside Networks, we focused our efforts on the distance education market. Along the way, we received calls from other companies asking, “Don’t you realize that BigBlueButton could be used in market X?" (An example of X would be remote health care.) From a business point of view, they would be correct. At the core of BigBlueButton is the ability to share voice, video, desktops, slides, and chat – these are all features that can be applied to many different markets. But from an entrepreneurial point of view, we adopted Geoffrey A. Moore's “crossing the chasm” strategy (http://wikipedia.org/wiki/Crossing_the_Chasm) and focused on one market to generate awareness and word-of-mouth marketing. Furthermore, our absence in other markets created opportunities for other companies, which we believe contributed to the health of the ecosystem and has lead to partnerships opportunities for Blindside Networks.

Lesson 2: Provide first-class community support for the open source project. This appears to be counter-intuitive: how can a company provide commercial support when its developers (wearing their open source hats) are providing first-class community support? We have seen other companies do it differently: they state upfront that they provide no support in the mailing lists, and if you want their support, you must pay. We believe that adoption of any open source software begins with trial testing and usage. Without a successful trial, there can be no large-scale deployment. Because BigBlueButton is free/libre open source software, it is very easy for a university or college to start a trial *if* they can get it working properly. We take the perspective that, if we consciously commit a portion of our resources to assisting others on the mailing list, then as their adoption of BigBlueButton grows so does the pool of potential cus-

tomers that may approach us later on. As an analogy, if you want a bountiful harvest of oranges, you have to be willing to plant many seeds and nurture many trees before they will bear fruit.

Lesson 3: Be upfront with your business model. People buy from people, but they will only buy if you let them know what you sell. When potential customers approach us, they are looking to accelerate their deployment of BigBlueButton to end users and reduce their risks of using open source. We let them know upfront that: i) there is a business model behind the BigBlueButton project; ii) Blindside Networks earns its revenues through support and services; and iii) our revenues are funding the development of BigBlueButton for the long-term. We charge a premium for our support, but the message we try to convey is that you want to pay a premium – and you want to know that others are also paying a premium – to be confident that we have the means to accelerate improvements to BigBlueButton for your benefit.

Lesson 4: Be clear about what you will not do. Early on in the mailing list, we received a lot of requests to help change the brand of BigBlueButton, which usually meant changing the interface to the point where it no longer had any references to BigBlueButton. After a while, we publicly stated that we would no longer volunteer our time for such efforts, explaining that it was tantamount to asking the community to volunteer their time with no benefit to the community itself. While a company could still internally rebrand BigBlueButton itself, given enough time and effort, we point out it would be more cost effective to engage commercial support from other companies in the ecosystem. In this way, companies in the ecosystem earn revenue which, in turn, supports the development of BigBlueButton.

Lesson 5: Hire a designer. There is a story from the early days of Google that the founders lacked the design skills to create a fancy home page, so they left it simple. After a while, that simplicity became part of their brand. In some ways, the same occurred with Blindside Networks: none of the co-founders were graphic or industrial designers, so we focused on improving the technical aspects of BigBlueButton and left our websites (and BigBlueButton) with a very simple interface. The author personally believes the best mix of co-founders is a group that draws from three skill sets: developer skills, sales skills, and design skills. When we look at other companies such as Heroku (http://heroku.com) and GitHub (https://github.com), it is obvious that they have strong designers in the company. Just as you cannot code your way to sales (i.e., you need someone to ask for the money), you cannot code your way to a good user interface design (i.e., you need to have some in-house design skills). We are growing our design resources now, but had we hired a full-time designer early on we could have established a stronger visual brand for both Blindside Networks and BigBlueButton.

Lesson 6: Ask qualifying questions to determine if the prospect has experience with open source. As much as we wish it were true, not every individual who calls or emails our company for support will become a customer. The challenge is to figure out which ones will. To help assess which prospects are willing to pay for services around open source, we (eventually) learned to ask the following key questions:

1. Have you worked with suppliers of other open source software before?

2. What is your business model for using BigBlueButton?

3. What is your daily rate for professional services?

The first question reveals whether they have ever paid for support for open source software. If the answer is "no", then we ask additional questions to figure out if they equate "free" open source software with free (or low-cost) support. In most cases, they are seeking low-cost support because

there is no business model behind their use of BigBlueButton. They are simply consumers of open source software. As such, we encourage them to engage the community for support, pointing out that the more they contribute to the community (in the true spirit of open source communities), the more support they get back.

If there is a business model, then we explore whether it is based on cost reduction or revenue generation. Cost reduction is good because most organizations can justify spending money in the short term to save money in the long term. When it is revenue generation, we ask if they intend to generate new revenue from BigBlueButton or incorporate it into an existing product and make revenues later on from increasing their customer base. In the former, there are opportunities for sharing revenue and growing together, but in the latter, any additional support costs for BigBlueButton will likely be viewed as another cost of doing business, and will be more scrutinized and reviewed.

The third question helps determine whether their rate for professional services is in alignment with ours. If the gap is too large, such as when doing business with companies from India, the likelihood that they will pay for our services is low. We are not surprised by this, and, in such cases, we encourage them to focus their resources on generating new revenues with BigBlueButton. This reduces their risk of spending what is perceived to them as large amounts of money without a certain return on their investment, and this positions Blindside Networks as ready to assist when there is a business case to justify acceleration of growth.

Lessons Learned Leading an Open Source Project

Lesson 1: Treat each release as if it were a product release. As open source developers, a perennial question we faced for each release was: "Have we done enough testing?" Fortunately, Blindside Networks had paying customers right from the beginning, so we established the mindset early on that the recipients of each release were paying customers, not just other open source developers. If it were only developers, we could cut some corners and work on new features. Instead, we spent an average of four extra weeks near the end of each release cycle fixing countless small issues that might be invisible to a developer, but not to a customer using BigBlueButton for a three-hour lecture. This investment in testing meant that we delivered fewer features in an iteration, but it also meant we had fewer issues to patch after a release, and our product was viewed as more solid. Ultimately, this level of polish has lead to a wider adoption of BigBlueButton, which has lead to more support and service opportunities.

Lesson 2: Put on your developer hat when communicating with the community. We are running a business, but when interacting with the open source community, we put on our developer hats and treat other members as peers, not prospects. We have seen mailing lists of other open source projects degrade into a forum where vendors rush to posts unhelpful answers with the tag line: "Contact me off list for help." Once a community reaches that point, most newcomers quickly conclude that the smart people have all left, one's contributions will not be reciprocated, and all that remains are the vultures picking over the bones. BigBlueButton has three mailing lists: developer, setup, and users, and we take the perspective that we are not there to sell, but to solve problems. As a result, there is a very healthy exchange of ideas and support on the mailing lists that strengthens the community and encourages others to reciprocate.

Lesson 3: Figure out the licensing model early on. All of the code written for BigBlueButton is open sourced under the Lesser GNU Public License (LGPL; http://gnu.org/licenses/lgpl.html), but it did not start out that way. Early on, we integrated another open source project called Xuggler (http://www.xuggle.com/xuggler/), which, at the time, was licensed under the GNU Affero

Public License (AGPL; http://gnu.org/licenses/agpl.html). The AGPL, unlike the LGLPL, is a reciprocal license and requires a company to make available under an AGPL license any code linked with Xuggler, even when offering a hosted product. After incorporating Xuggler, we received a friendly-but-firm call from the Xuggler developers stating that if we intended to use Xuggler, then we must open source BigBlueButton under the AGPL license as well. To avoid having to open source all of BigBlueButton under AGPL, we isolated the desktop sharing component as a module so that BigBlueButton did not depend on desktop sharing. Hence, we only needed to license the desktop sharing component under the AGPL. To make it possible for companies to integrate BigBlueButton into their product, we offered a dual-license for the desktop sharing component: AGPL and commercial. This was all ad hoc. While we did make some sales from a commercial license for desktop sharing, the terms of the AGPL license were restricting others from integrating BigBlueButton into their open source projects. After a few months, we reworked BigBlueButton so it did not require Xuggler and reverted our codebase to LGPL, betting that by accelerating the adoption of BigBlueButton we could earn more revenues in the long term. While it is impossible to pursue both strategies in parallel to determine if this is true, we believe it so. (As an interesting side note, the Xuggler project eventually moved their codebase to LGPL as well).

Lesson 4: Write a list of frequently asked questions (FAQs) from the beginning. The total number of posts in our mailing lists now exceeds eleven thousand. Early on, we saw the high traffic as an endorsement of BigBlueButton; we viewed each question as an opportunity to build a relationship with a newcomer and demonstrate that the members of the BigBlueButton project really cared about ensuring they had a positive experience with the project. Looking back at some of those early threads, such as setting up BigBlueButton behind a firewall, there were over thirty messages of patient support. When the member finally got their BigBlueButton server configured, they were very happy, but it took a lot of effort on our part to achieve it. Now with a FAQ of over 100 answers, we still, for example, answer lots of questions around setting up BigBlueButton behind a firewall, but the threads are shorter, the effort is less, and the members are still just as happy when their installation of BigBlueButton works.

Conclusion

Some of our strategies – such as offering first-class community support in the mailing lists – seem counter-intuitive, but our experience over the past three years suggests that, when an education or commercial institution makes a decision to deploy BigBlueButton, they are more likely to purchase from someone who has invested their expertise in providing them support long before deployment was even considered.

We thought a lot about how to build a strong community. It boils down to this: whatever behaviour we expected of others (professionalism, reciprocity, and non-solicitation), we had to exhibit it ourselves.

We have planted a lot of seeds with BigBlueButton, and our strategy to focus on a single market segment has created opportunities for other companies, which is good for the ecosystem. An ecosystem with only one company is not a healthy ecosystem, and we do not want to be the only company offering commercial support for BigBlueButton, we just want to be the best.

Fred Dixon is CEO of Blindside Networks. He is a serial entrepreneur, having been CEO of two previous companies: Databeacon (acquired by Cognos in 2004) and OpenLava software. In 2003, he was selected as one of Ottawa's "Top 40 under 40" executives by the Ottawa Business Journal. In 1992, he earned a Bachelor of Mathematics from the University of Waterloo. He proudly wears his developer hat when communicating with other members in the BigBlueButton community.

Developing Silicon IP with Open Source Tools

Arthur Low

> *"The complexity for minimum component costs has increased at a rate of roughly a factor of two per year... [T]here is no reason to believe it will not remain nearly constant for at least 10 years... Perhaps newly devised design automation procedures could translate from logic diagram to technological realization without any special engineering."*
>
> Gordon E. Moore
> *Electronics,* April 19, 1965

The electronic design automation (EDA) tool industry is big business, and commercial licenses are extremely expensive. Open standards have driven many proprietary EDA technologies to be publicly released as free/libre open source software (F/LOSS) and some have become IEEE standards. Competition has partly given way to collaboration and has led to these standards. The development path of important EDA tools frequently now employs F/LOSS practices, which have overcome resistance to collaborative innovation between competing businesses. F/LOSS technologies are at the vanguard of leading-edge system-on-chip (SoC) design, not just because they are free, but also because they are valuable.

The first commercial integrated circuits (ICs), designed by hand, helped guide manned space flight to the moon on the Apollo missions. In the past decade, silicon IP firms have shown they are limited only by their ideas, not by limited investment opportunities, and SoC firms have shown they can greatly reduce costs while innovating on the development of the largest new IC designs. This high-end technology is made accessible to startups because of open source. It is no longer just for mega-corporations.

This article reviews the history of key advances in ICs and EDA tools. The common theme presented in this article for the driver of technology innovation is the requirement to develop the most advanced microprocessor possible. Today, a low-cost, high-value-added business model can efficiently serve the market for IC subsystems licensed as intellectual property (silicon IP) in the form of compilable source code. Alternatively, for larger SoC designs, engineering budgets can be shifted from the purchase of a relatively small number of high-cost EDA tool licenses to open source EDA technologies that can be run on massive compute-server farms. The two business models are not theoretical, but realistic. The author explains how his company (Crack Semiconductor) developed commercially successful cryptographic silicon IP using entirely open source EDA technologies and how another company (SiCortex) pushed the limits of IC design and open source EDA tools by simulating and verifying a massively parallel supercomputer.

Developing Silicon IP with Open Source Tools

Arthur Low

Introduction

In 1965, Gordon E. Moore was Director of Fairchild Semiconductor's R&D laboratories and made his famous observation, as quoted above, that would become known as "Moore's Law". At that time, ICs were closely coupled to rocket science (figuratively and literally), and microprocessors had not yet been invented. Fairchild was then busy supplying the first commercial ICs for the Apollo Guidance Computer. This computer was used on the successful moon missions, and it helped bring Apollo 13 back home. Thousands of Fairchild ICs, containing only a NOR gate with three transistors each, were used to build one system. It would take more than twenty years and billions of dollars worth of industrial R&D to achieve Moore's prediction of EDA tools that could "translate from *logic diagrams* to technological realization without any special engineering".

In 1968, Gordon Moore and Robert Noyce founded Intel, a classic Silicon Valley startup, which in 1971 produced the first microprocessor (uP) design (http://intel4004.com). This first microprocessor contained 2300 transistors. Intel uP transistor counts grew to 820 million in 35 years, a 2.07 times increase every two years. Moore's Law was coined by Carver Mead, professor of VLSI design at Caltech (http://tinyurl.com/3tualzb), in reference to Moore's prediction of the growth rate of IC designs in his now-famous 1965 *Electronics* article (http://tinyurl.com/yhab3vp).

By the mid-1980s, a more efficient alternative to logic diagrams appeared: software-like hardware description languages (HDLs) that model concurrent logic circuit activity. The US Air force required the complete logic functional description of a digital IC. This requirement led to the open language specification called VHDL. Proprietary logic simulation technology helped fuel the growth of the commercial IC industry. As a business strategy to counter VHDL's gains in market share, the dominant language (Verilog; http://www.verilog.com) was later released to an independent organization so it could be developed as an open specification. Ironically, both VHDL and Verilog have become IEEE standards managed by the same organization, now called Accellera (http://accellera.org).

Proprietary logic design and simulation tool projects, for almost 50 years, have been supported within vertically integrated mega-corporations, and they, like the tools, have prospered and withered. Several EDA and IC startups flourished to become mega-corporations themselves, but many failed and quietly faded away. IC companies (such as Intel), telecommunications R&D labs (such as Bell-Northern Research, or BNR), and computer giants (such as IBM and Digital Equipment Corporation, or DEC), all developed internal EDA tools to support their own chip designs. Later, a robust EDA industry formed as engineers left large companies to start new firms.

In the next section, we discuss the emergence of the silicon IP business model and how proprietary tools have given some way to F/LOSS tools. In the last section, we show that F/LOSS tools can help silicon IP business founders avoid dilution of their shares in the company and loss of control before the idea is turned into an important innovation. They can also support the economical trade-off of expensive tool licenses versus computing resources. An example of each is given: F/LOSS tools enabled the author to develop commercially successful cryptographic silicon IP, and the extreme limits of IC design were pushed by simulating and verifying a massively parallel supercomputer.

The Emergence of the Silicon IP Business Model

In this section, the development of the microprocessor is used as the common design element. The silicon IP business model is shown to have been preceded first by the vertically organized, fabricating semiconductor company. The fabless

semiconductor company outsourced IC fabrication and so required vastly smaller investment capital, but each business type still designed everything "in-house". Design teams were large and significant financial resources required venture capital to be raised. At a critical moment, design tools enabled highly productive and much smaller design teams, and the microprocessor concept took a leap forward with the introduction of the reduced instruction set compute (RISC) as a core of silicon IP that was licensed without ever having been manufactured. This was followed by the introduction of the programmable logic chip, and finally, F/LOSS design tools that enable silicon IP startups to design, test, and deliver working IC designs with almost no capital investment beyond their intellectual contributions.

Long before the silicon IP business model emerged to fill the current market need, microprocessors were designed by very large high-technology businesses. These companies were often vertically integrated by necessity. This led to the development of internal EDA technologies that predated equivalent commercial offerings. An advanced technology company owned its own IC design and manufacturing processes, and its internal tool development programs were closely coupled to those processes. Examples include IBM's "Einstimer" tool for checking if the chip signals meet timing requirements and BNR's "Funsim" hardware design simulator.

The vertically integrated "semiconductor fabrication-oriented" business model was joined by the "fabless" semiconductor startup, which was enabled by the availability of commercial EDA tools. The fabless semiconductor company – clearly viable in the mid-1990s – now could design all the logic functions required in an application-specific IC (ASIC), without needing a manufacturing capability. The ASIC design could then be manufactured by a semiconductor foundry, called the "fab". But, relentlessly, IC designs have grown by Moore's Law. It is now quite impractical for one company to design all the logic functions in the chip, so the silicon IP market now supplies a significant ratio of the functional logic to a company that integrates these subsystems on an SoC.

The classic IC microprocessor that was designed and manufactured by one company started to face competition in the 1990s. ARM Ltd., which started as Acorn Computers and was joined later by Apple and VLSI Technologies, developed the silicon IP business model by introducing a small, but powerful reduced instruction set computer (RISC) design to be licensed to other companies that would embed the silicon IP into their IC design. Embedded RISC processors are used in virtually all of today's hot products, such as smart phones and tablets. By 2008, over 10 billion ARM processors were licensed. ARM's initial success as a silicon IP vendor in the microprocessor market demonstrated that massive industrial resources are not required for the silicon IP business model. ARM proved that small design teams – even one individual – can produce valuable processor designs as silicon IP.

A final plank in the silicon IP business model platform is the field programmable gate array (FPGA). The FPGA has allowed Moore's prophetic call for "design automation procedures [that] could translate from logic diagram to technological realization without any special engineering" to be realized. The two leading FPGA vendors, Xilinx and Altera, provide low-cost or free tools to automatically convert HDL source code to a device-specific technology, place it on the pre-manufactured chip, and wire up the components in minutes.

From Proprietary EDA Tools to F/LOSS Solutions

Initially, silicon IP vendors had little option but to acquire expensive EDA tool licenses where each simulator might cost, for example, $25,000. This need usually forced the founders to give up

equity to fund their innovations. Now with F/LOSS tools for EDA, innovators can develop their advanced silicon IP, and with FPGA technology, the silicon IP vendor can let a potential customer evaluate the IP in their own design for very low cost and risk.

The emergence of an F/LOSS suite of EDA tools for the front-end logic design and verification has followed an interesting path. In this section, the requirement for a technology that would allow software and hardware to be modeled and simulated is described. Two problems drove this movement to open source EDA tools; one is technical and the other is business-oriented, concerning how competitors can collaborate. As a secondary effect, open source technologies have enabled the low-cost silicon IP startup as a viable business to supply both real designs and models to enable faster and higher-quality solutions. These technologies have liberated the silicon IP business model from its dependence on highly diluting venture capital.

The technical problem to be solved can be summarized by the following question: How can complex microprocessor designs execute software, given that the interaction between software and hardware must be well understood before the design is committed to silicon? This concept is called co-simulation. The competitor's problem was to determine how the business can profit if it shares its advanced EDA tool innovations with its competitor.

In the 1990s, the need for a co-simulation technology lead DEC to develop Verilator – discussed in greater detail below – purely for DEC's internal use. DEC's misfortune has lead to other's fortune, because Verilator is now the leading F/LOSS tool for the silicon IP startup on a microbudget. The demands for even higher system-level co-simulation and modeling technology led to SystemC as a collaborative effort between EDA companies. Verilator and SystemC form the "killer app" for silicon IP startups.

Extreme engineering challenges confront leading-edge IC development projects. Internal innovation is often the only option to overcome a technical limitation of a commercial EDA tool. However, with access to the source code, a feature can be added or a bug can be fixed directly.

The microprocessor IC is typically so complex that many simulations of its operation are required to verify the design. Requirements led to capabilities that frequently could only be developed by internal EDA tool development teams working closely with the processor designers. Most vertically integrated companies funded (and many continue to fund) internal EDA tool development and had proprietary design flows. Industry standardization was regarded by leading-edge companies, frequently correctly, as imposing a step down in capability. Processor designs were often far more advanced than the logic devices that could be designed and manufactured with commercial EDA tools, and they often pushed those tools past the breaking point. Arguably, the internal tools of Intel, IBM, DEC, and BNR were the crown jewels of each corporation. But this common practice of internal development also led to wasteful duplication and resistance to external ideas and innovations, which were ignored because the were "not invented here".

Many companies today are choosing to collaborate with their competition on the development of fundamental technologies by supporting F/LOSS EDA initiatives. However, this is not so in the case of DEC against Intel.

Intel is now a dominant microprocessor IC company, but many companies vied for the position, including IBM, AMD, and DEC. Intel only emerged as the dominant processor vendor after it began its “Wintel” collaboration with Microsoft. DEC's innovation for its Alpha processor is a good case study. To verify the Alpha, DEC developed the tool called Verilator starting in 1994. The requirement was to co-simulate C (software

code) and Verilog (hardware code) together. Verilog was "verilated" to C for DEC's Alpha uP project and then compiled with a C compiler. In 1998, nearing the end of a long run, DEC publicly released the source code for Verilator before the company was sold to Compaq. Since 2001, Verilator has been maintained by Wilson Snyder (http://www.veripool.org).

Processor design is now more complex than ever, and silicon IP cannot be developed following the writing of detailed design specifications for both software and hardware. This takes too long and correct designs emerge from frequent, short iteration cycles using first models and then more detailed modules. So, leading edge EDA companies decided to collaborate by forming the Open SystemC Initiative (http://systemc.org). By collaborating in the specification of the SystemC language and its later extensions, these companies assured that a stable market for their value-added co-simulation and modeling design tools would exist.

Silicon IP Developed With Verilator and SystemC

The author's company, Crack Semiconductor (http://cracksemi.com), recently licensed to a large European client an RSA (http://wikipedia.org/wiki/RSA) public key cryptographic security processor to accelerate banking security transactions. Crack Semiconductor's RSA processor optimally multiplies numbers that are enormous: 1024-bits and larger. The IP was developed in Verilog and was verilated to SystemC, and then was compiled with G++ for simulation. A SystemC test environment generated random numbers, which were used as a basis for an external function call to pre-compute system constants and the expected results using GNU bc, an arbitrary precision calculator language. The SystemC test then read the expected results from a file, programmed the virtual processor, and executed the simulation of the verilated RSA processor. During development, when results were incorrect, bc scripts were written to compute intermediate results to compare against the values generated by the 32-bit multiplier, or at any other selected observation point in the processor. There are no commercial tools available to do this kind of "specialized engineering". Finally, for delivery to the client, free synthesis tools offered by Xilinx were used to convert the soft IP core to a technology-specific format used by the client.

Technology innovation in IC design methodologies often triggers new ideas for F/LOSS EDA tools that will support the next generation of IC designs. Wilson Snyder was a member of the development team that designed the 972-node parallel SiCortex supercomputer. The 200 million transistor SiCortex chip contains 64-bit RISC processors that represents one "node" in the supercomputer, and was developed using the same basic technology used by Crack Semiconductor. The SiCortex team exploited all manner of open source technologies to enable up to four hundred Linux compute servers to run simulations in parallel and report what aspects of the test plan have been covered. This new open source technology is called CovVise (http://veripool.org/wiki/covvise). CovVise is not just one technology, but actually leverages a wide variety of F/LOSS technologies generally referred to as LAMP (Linux operating system, Apache web server, MySQL database, and Perl/PHP scripting language). A commercial Verilog simulator license has a list price around $25,000, so economically, Verilator, SystemC, and CovVise represents a compelling solution for large IC design teams (http://tinyurl.com/3nea5hz).

Conclusion

On the 46th anniversary of Gordon E. Moore's seminal paper in Electronic, F/LOSS EDA tools enable logic descriptions to be technologically realized without any special engineering, as Moore predicted. Front-end design engineering of silicon IP still requires extremely specialized engineering problem-solving efforts, and usually commercial tools do not fully address the problems that arise. Engineers at the cutting edge of technological development routinely must invent new design and verification tools, and F/LOSS is indispensable in this effort.

All manner of business models are receptive to F/LOSS EDA tools. From silicon IP startups with micro-budgets to large corporations like NXP (Philips Semiconductor), many companies use these tools today. They do save money by using F/LOSS, but they do not use them just because they are free of monetary cost. They use them because they are valuable. From DEC to SiCortex and Crack Semiconductor, open source Verilator, combined with IEEE Standard SystemC and SystemPerl, offers compelling value for silicon IP startups.

Arthur Low is the founder and Chief Technology Officer of Crack Semiconductor, a supplier of high-performance cryptographic silicon IP used in some of the most demanding security applications. Arthur has worked for a number of IC startups as a Senior IC designer and Architect, and gained much of his fundamental IC design experience with Bell-Northern Research in the early 1990s and with IBM Microelectronics in the late 1990s. Arthur has a BSc. degree in Electrical Engineering from the University of Alberta and is completing his MSc. degree in Technology Innovation Management in the Department of Systems and Computer Engineering at Carleton University.

Proving the Expertise of Freelance Android Developers

Igor Sales and Aparna Shanker

"Believe one who has proved it. Believe an expert."
Virgil

Android continues to grow in popularity as a mobile operating system. With this constant growth in popularity comes a demand for skilled, specialized platform and application developers rather than just generalists that are currently provided by outsourcing firms. This article describes a business opportunity where a collective of proven experts can be used to fulfill this need for specialized developers. A key aspect of the proposed solution is the provision of proven expertise by certifying and guaranteeing the level of specialization of developers in the collective.

Introduction

The Android operating system (http://android.com) is the fastest-growing operating system in the mobile space (http://tinyurl.com/4zov69c) and has even surpassed the iPhone in market share (http://tinyurl.com/2vnjron). A key reason for Android's growth is that manufacturers have recognized the benefits of this operating system. With Android, mobile phone manufacturers do not have to spend resources to create and maintain their own version of a mobile operating system. Another reason for the success of Android is its sponsorship by Google, which through its sheer market power provides significant financial support and credibility.

Android is also popular with developers because it makes development easy. It uses a well-known programming language (Java) and there are no prerequisite membership barriers to its development environment. However, it is arguable that Android's greatest strength is that it is open source. Developers from all corners of the world have open access to the full source code. Even though the majority of contributions to the Android operating system come from Google's Android team, the body of Android developers outside of Google produces an ever-growing quantity of Android software, turning Android into a very attractive platform for mobile consumers and producers.

Although Android is uniquely positioned and has a competitive edge over rival, proprietary mobile operating systems, there remain important challenges faced by Android developers and the software development firms that employ them. In this article, we describe these challenges and then propose a solution that we intend to develop as a business opportunity. Our solution changes the current environment by offering a match-making platform that brings together a collective of specialized Android developers and software firms. We will demonstrate that the power of a collective is more compelling than the current solutions available to individual contractors and outsourcing firms.

Proving the Expertise of Freelance Android Developers

Igor Sales and Aparna Shanker

The Problem

A firm typically bases a hiring decision on a developer's resumé, portfolio, and interview. Even still, a firm cannot know for certain if the developer has the skills necessary and will perform effectively. Equally, the developer does not know if the firm will be a good fit for their needs. Accordingly, many firms hire new developers only if one of their own developers vouches for the candidate, who also has a source of inside knowledge through which to assess the job on offer. While this strategy is perceived to work well (since the firm and the developer rely on the trust relationships between the firm, the established developer, and the candidate), reality is less clear. It can be difficult for firm's to assess whether a recommendation is based on friendship, skill set, past performance, or other factors. For example, many firms offer bonuses to established developers as an incentive to recommend candidates from their networks, which may in fact complicate the situation. There remains a need for firms to accurately assess a candidate's skills and there remains a need for developers to prove their expertise.

In the context of Android, an additional challenge is that many freelance developers do not wish to work exclusively or permanently for a particular firm. In fact, many wish to be strongly associated with the Android project, rather than with any firm using it. In some cases, this comes from an affinity for the principles of free/libre open source software, including a desire to see their contributions remain in the public domain. The incentive to contribute to open source might be for future, if not immediate, monetary gain (Lerner and Tirole, 2002; http://tinyurl.com/4ymdg6s). These freelance developers offer their development services for hire, but the challenge for firms is to determine which freelancers would be a good match to work for their particular firm.

Generally, firms have three options:

1. Hire developers as permanent employees.

2. Hire freelance developers on temporary contracts.

3. Hire an outsourcing firm that employs teams of developers.

Firms gain flexibility by outsourcing or hiring freelancers on contract, however they may lose the benefit of having a consistent team of developers that work together regularly. Of course, the options above are not mutually exclusive. Firms may supplement a permanent team of developers with freelancing and outsourcing solutions.

A challenge with outsourcing development work is that much of the work will be assigned to generalist developers, who may require additional training or management oversight to meet the project's requirements. Large outsourcing firms provide a good solution for well-defined projects, where the task breakdown is deterministic and project management is only a matter of execution. Most firms building on top of the Android platform (i.e., changing the operating system source code to suit their particular needs) do not match this profile because the Android platform is still undergoing many major changes, therefore very hard to create predictable execution plans for these projects.

Our Solution

The authors are in the early stages of developing a solution to this problem as a business opportunity. We intend to develop a network of Android freelancers, project managers, planners, and software manufacturing firms that base their products and services on Android. This network, which we call the Android Freelancer Net-

work, will function as a collective that connects software development projects to the most suitable developer or team of developers.

The key aspect of this solution is its ability to prove the expertise of the workers with respect to Android. This proof is provided through visible referrals, certification programs, commit history, and other measures of expertise and past performance. The platform will analyze the contributions to the Android platform, both for individuals and firms, and maintain a database to track their performance. The platform will also provide details of members' interactions and feedback, along with lists of skills required to perform certain tasks with Android.

Our business will sell memberships to this network, which will connect parties together in a way that demonstrates their level of expertise and track record with other members. Further, we will provide project planning and maintenance services to help members that might have a detailed plan on how their solution should be built and maintained. Compared to outsourcing firms providing generalized expertise, the unique feature of the Android Freelancer Network is its ability to provide the best matches between freelancers and firms based on the developer's proven expertise and the firm's requirements.

For firms, this solution will reduce a project's costs and risks by finding the right people to produce and execute a project plan given the firms' requirements. Firms can be confident that their product requirements are going to be met since they are matched with experts that can prove they have worked on areas of Android defined by the firm's requirements. Freelancers with proven expertise are more likely to design solutions that match the firm's requirements than generalists or developers whose experience is unproven.

For developers, this solution will reduce their search costs and help them find the best fit for their skills. The ability to prove their expertise is a unique and proprietary method to analyze their prior contributions and reputation with the Android platform. Also, developers will only be subject to membership dues if they find jobs through the network.

Conclusion

In this article, we described the attractive features and challenges of the Android operating system as a complementary asset to mobile software and hardware manufacturers. The high demand for Android developers has created a market where generalists can pose as Android experts, which is a problem for both freelancers and the firms looking to hire them. We outlined our business opportunity, the Android Freelancer Network, which will match developers and firms based on their expertise, past performance, and current requirements. The power of this collective is its ability to attract and maintain a specialized pool of professionals with proven Android expertise. This solution addresses problems facing both developers and firm and will contribute to the growth of the Android operating system.

Proving the Expertise of Freelance Android Developers

Igor Sales and Aparna Shanker

Igor Sales is a freelance software developer and entrepreneur in Ottawa, Ontario. He holds a Master's degree in Software Engineering from the University of Ottawa and is currently a graduate student in the Technology Innovation Management program at Carleton University. Igor has coded device drivers, frameworks, and applications as well as interfaced directly with customers to solve their technical challenges. He is now focusing on mobile software frameworks and applications.

Aparna Shanker is a network engineer with Alcatel-Lucent in Ottawa, Ontario. She is currently a graduate student in the Technology Innovation Management program at Carleton University and holds an undergraduate degree in Computer Engineering from Queen's University, Kingston.

Upcoming Events

May 11 - 14

BSDCan 2011

Ottawa, ON

"BSDCan is a developers conference with a strong focus on emerging technologies, research projects, and works in progress. It also features Userland infrastructure projects and invites contributions from both free software developers and those from commercial vendors."

http://www.bsdcan.org/2011/

May 25 - 26

mesh: Canada's Web Conference

Toronto, ON

"Canada's leading online conference, mesh explores how the Internet is changing how we live, work and play. mesh is divided into four streams – media, society, business and marketing – to provide an overall of the key trends, issues, companies and tools. mesh is designed to be interactive and engaging with attendees being as much of the programming as speakers. With a few exceptions, mesh is a slide deck free event that encourages people to get involved, network and, well, mesh."

http://www.meshconference.com/

June 6 - 10

Net Change Week

Toronto, ON

"Net Change Week (NCW) is Canada's premier event on social tech for social change. The weeklong series of events features training workshops, evening programming with guest speakers, lab sessions and plenty of opportunity for networking. In its third year, Net Change continues to be committed to digital literacy and pushing the boundaries of technology's potential to yield greater impact."

http://netchangeweek.ca/

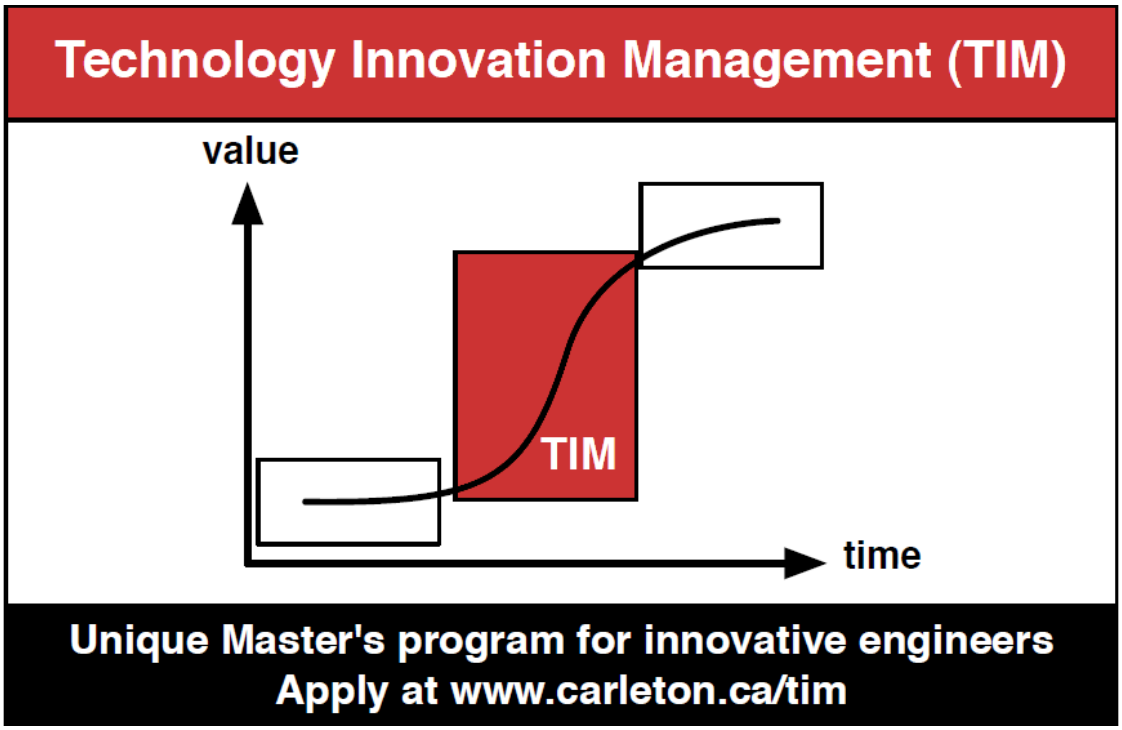
June 13 - 15

Ottawa Linux Symposium

Ottawa, ON

"The Linux Symposium has been an annual gathering of Linux and Free Software developers, professionals, and enthusiasts since 1999. We strive to be good community members and to provide a neutral environment and encourage open discussion."

http://www.linuxsymposium.org/2011/



TIM is a unique Master's program for innovative engineers that focuses on creating wealth at the early stages of company or opportunity life cycles. It is offered by Carleton University's Department of Systems and Computer Engineering. The program provides benefits to aspiring entrepreneurs, engineers seeking more senior leadership roles in their companies, and engineers building credentials and expertise for their next career move.

Carleton UNIVERSITY

Contribute

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 to any of these questions is "yes," then 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. Formal first-person style (we only) may also be acceptable.

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

June 2011: Technology Entrepreneurship II

July 2011: Women Entrepreneurs

Contribute

Formatting Guidelines:

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.