# OSBR.CA

The Open Source Business Resource



MAY 2010

# MAY 2010

**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 this*** issue of the OSBR is Communications Enabled Applications (CEA). While using software to enhance communications is not new, there remain many untapped business opportunities available to the savvy entrepreneur as well as opportunities for any organization to improve the relationship with their customers. The authors in this issue draw upon their experiences to show the benefit in CEA and offer practical examples for those wishing to tap into this powerful resource.

***This issue includes articles from*** six authors, of which three work for multinationals, one works for a small company, and two are founders of innovative technology companies.

***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.

***The editorial theme for the*** upcoming June issue of the OSBR is Growing Business and the guest editor will be Mekki MacAulay. Submissions are due by May 20--contact the Editor if you are interested in a submission.

**Dru Lavigne**

**Editor-in-Chief**

*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 BSD Magazine and is the author of the books BSD Hacks, The Best of FreeBSD Basics, and the Definitive Guide to PC-BSD.*

***As anyone with a smartphone*** can appreciate, the power of having a mobile phone that can can access the Internet creates significant value for users and opportunities for businesses. This device-level integration of communication services and web applications is now common. However, we are only now scratching the surface of the next step in value: application-level integration. The ability to integrate communications services within web applications opens up tremendous opportunities. Examples of communications enabled applications include simple click-to-call links on a website, conference calls initiated by applications or users in response to events, interactive voice response menus, and any number of other ways that communication services, such as messaging, voice, and conference calls, can be integrated into an application to add value.

***I recently had the pleasure*** of coordinating the activities of a group of entrepreneurs, developers, and architects as they explored together how CEAs can add value to their offerings and their business ecosystem. The Elena Project was funded by IRAP (http://nrc-cnrc.gc.ca/eng/ibp/irap.html) to stimulate small technology companies to develop working prototypes of CEAs and expand the capabilities of the Coral CEA business ecosystem and sandbox (http://coralcea.ca). The project focused on using four voice services and the open source web conferencing tool BigBlueButton (http://bigbluebutton.org).

***Among the outcomes of this*** project was the realization that significant value can be leveraged when communications features become integral parts of applications. In this issue of the OSBR, a diverse group of authors share their experiences and knowledge to help others explore the value CEAs could bring to their own offerings. All of the authors in this issue participated directly in the Elena project or present analysis relating to Coral CEA.

***Craik Pyke examines the history*** of telecommunications programmatic interfaces and their accompanying business models. He explores the shift from business models based on monetising application programming interfaces (APIs) directly, to models based on monetising the communications capabilities of the applications that use these APIs. Finally, he outlines an ecosystem approach to leveraging CEA services.

***Daniel Cardenas shares his practical*** experiences integrating communications services into an event-management system. His analysis of the business case and technical approach illustrates how enabling communications yielded immediate improvements to operations, customer service, and revenues. He offers specific recommendations for others who may wish to follow a similar approach.

***Andrew Ceponkus explores the role*** open source software and CEAs play in the future of patient record management and telehealth services. He identifies an opportunity for entrepreneurs who are open to collaboration and partnership in a space that has been traditionally dominated by large projects and large companies.

***Patrick O'Halloran shifts the focus*** from customer needs and technology to the intellectual property and licensing issues that should not be overlooked, particularly by the consumers and providers of CEA platforms. After providing some necessary background, Patrick provides examples of how to address the implications of dependencies inherited from the use of CEA building blocks beyond a sandbox environment.

***Jean-Pierre Poulin explains how his*** customers' reactions to phone features convinced him to give CEAs a closer look. He outlines the benefits of integrating telephony features into an application, describes the necessary conditions for this to be successful, and shares practical tips to help others overcome obstacles.

***Elias Majic recently integrated voice*** and web conference services into an open source customer relationship management (CRM) system. He describes the key features of existing CRM systems and the increased value that can be achieved through communications enablement. His insights into the required technology choices and the general lessons he learned along the way are applicable beyond CRM systems.

**Chris McPhee**

**Guest Editor**

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

*"If you are the person in your company trying to define the business case for an API to the executive team, there is a big hurdle to overcome, because business executives tend to see an API as a cost center and want to know how to measure the payoff."*

Laura Merling, VP Developer Platform & Programs at Alcatel-Lucent

Over the past decade there have been numerous attempts at opening telecom infrastructures to developers. As each attempt evolves to the next, there is an equal desire to monetise the exposure of telecom capabilities using traditional and well understood mechanisms: charge for necessary equipment upgrades and license the application programming interfaces (APIs) on a per-invocation or "block of simultaneous invocations" basis. However, the various vendors and development companies involved in creating applications with embedded communications capabilities have had to re-examine their business and technology models in an increasingly competitive applications market where the rate of applications failing to gain market traction far outweighs the rate of success.

This article looks at the history of telecommunications APIs and the predominant business models that have accompanied those interfaces. By analysing the history of telecom APIs and recognising the gradual shift from a strongly vendor controlled environment to a highly accessible component of information technology (IT) networks, we can recognise the shift in revenue generation from a typical monetisation model to a value based model. Additionally, we can examine how incumbents and new entrants are dealing with the more unpredictable business models and emerging methods for de-risking value based revenue opportunities.

**APIs as a Monetisation Mechanism**

Early programmatic interfaces such as computer-telephony interfaces (CTIs) permitted service providers and network equipment providers to develop and augment communications centric applications without waiting for the next revision of the communications system software. CTIs tended to be proprietary to the network equipment provider, creating difficulties for application and service providers to build capabilities that reached across the network equipment provider's infrastructures. Moreover, building developer skill-sets in cross-vendor APIs was challenging, leading to dependence on the equipment vendor's development services.

As a result, later efforts such as intelligent networking protocols were targeted at alleviating this dependance by focusing on heavily standardised APIs. These service provider targeted APIs tended to be targeted at telecommunications centric developers with in-depth knowledge of how communications systems function. However, the rigid standardisation of the APIs permitted building cross-vendor skills in organisations that were not controlled by the network equipment vendors. Similarly, enterprise communications equipment vendors adopted standardised interfaces such as computer-supported telecommunications applications at the behest of their customers.

The primary effects of this progressive opening of the communications system was two-fold:

- equipment vendors developed an additional revenue stream by charging for access to the APIs on their communications equipment

- companies specialising in application nodes were able to develop capabilities attractive to service providers and applicable across the multi-equipment vendor network

In many ways, the two outcomes of opening the communications networks became intertwined. Many network equipment providers such as Ericsson, Lucent, and Alcatel operated successful solution lines offering both service nodes and communications network interfaces. Such companies offered services and a coupled service creation environment on their service nodes, thereby controlling the use of the communications APIs to a known set of use cases. The enterprise communications environment similarly unfolded with Lucent, Cisco and Nortel offering APIs to their communications infrastructure and selling application and application creation environments leveraging the interfaces of their equipment.

For network equipment providers, the APIs became a monetisation mechanism. They either designed the applications or dealt with application developers that were direct customers. The business of enabling communications APIs became centred around how to generate the maximum revenue from the one-time sale of an application and the recurrent use of APIs by many applications.

Vendors who specialised in application nodes, such as Telcordia and Genesys, were at the mercy of the network communications provider both from the API implementation (whether the vendor elected to implement all of the standard or a subset) and from the API prices (a factor of both right-to-use licensing costs and hardware investment). They were forced to differentiate their applications from the network equipment provider's application while using the APIs implemented by those same providers.

Network equipment providers did not take objection to the application vendors, given that they generated far more revenue from API licenses than from application sales. Thus, even when not selling the application directly, the network equipment providers were capturing the largest share of revenue for each voice application deployed against their network equipment.

**The Emergence of Unified Communications**

This business model for voice services and voice related applications persisted throughout much of the digital switching and digital PBX (http://en.wikipedia.org/wiki/Pbx) era. Communication networks continued to evolve, giving rise to Voice over Packet technologies. These technologies caused users, providers and administrators to think about communications differently; no longer was the communications device a digital terminal attached to a closed copper loop. Instead, communications devices were becoming another computer accessible over the same network as other modes of communication and collaboration. As the voice communications device became simply another extension of the computer network, a new class of applications began to arise.

Voice over Packet gave rise to the class of integrated desktop applications now typically referred to as Unified Communications. While Unified Communications leverages Voice over Packet as an integrated component of the unified experience, many companies developed solutions to integrate legacy voice communications solutions. Key to integration was leveraging the APIs that had evolved as part of the digital equipment revolution, as well as leveraging Session Initiation Protocol (SIP) interfaces which had been added to the legacy equipment.

Given that the central focus of Unified Communications was to unite the variety of devices and communications mechanisms such as voice, email, instant messaging, video, presence, and mobility, there was no longer a tendency to rely upon the voice communications provider as the source of the applications. The most logical providers of Unified Communications were those vendors who were already part of the substantial desktop investment such as IBM and Microsoft. Many enterprises and service providers were able to leverage already purchased licenses for APIs into the voice communications systems for the new Unified Communications applications. Even when new licenses were required, the network equipment providers were no longer being engaged for new API functionality and were no longer able to demand premium prices for API licenses. Many of the network equipment vendors struck partnerships with the Unified Communications application vendors as a means to drive additional API license sales through sales of the existing API capabilities and incremental capabilities added to the API portfolios. The network equipment providers were able to retain a revenue stream based around monitising their API portfolios, though not as deep as that revenue stream had been in the past.

**Service Delivery Platforms**

The rise of Unified Communications began to signal a shift in the communications-centric applications development model. As the application distribution control shifted to more IT centric companies, the portion of application developers with deep knowledge of voice communications network functionality declined. Developers were versed in development models consistent with the desktop software they were integrating communication with, as well as Services Oriented Architecture (SOA) and Web 2.0 principles and methodologies. The APIs provided by the voice communications infrastructure did not lend well to these development models. Additionally, despite standards for many of the API formats, the disparity of implementation and the broad variety of available API standards had led to fragmentation in customers networks. As a result, service providers, enterprise administrators and application vendors began to leverage new platforms which both simplified the communications network APIs as well as provided a unifying translating gateway between applications and the communications networks.

Service Delivery Platforms (SDPs) are a combination of service creation environments, service execution environments, media control, and interface integration capabilities. They offer an integrated environment for developing and deploying applications. For application developers, SDPs provide a means to develop using telecommunications capabilities while avoiding complex APIs in favour of the simplified APIs provided by the delivery platforms. However, for service providers, enterprise IT administrators, network equipment vendors, and even SDP vendors, SDP is a difficult business case to rationalise against the existing business model.

SDPs are by design a middleware product. They consume APIs from the communications networks, consolidate, and re-publish APIs towards applications. In the model prior to SDPs, network equipment vendors licensed APIs while application vendors provided applications using those APIs. As the value model shifted to applications, particularly those offered as a service, the opportunity to monitise just the APIs diminished. The service model further pressured the telecom API business as new licenses could be acquired only as needed.

SDP vendors offered a consistent, consolidated platform for the creation and deployment of services with security, reliability and availability. Organizations requiring new functionality or compliance to new standards looked to the middleware vendors first.

Application development is not a guaranteed business. For every successful application that captures the attention of consumers and business users, there are dozens of failed applications. Capturing wallet-share with applications has increased in difficulty, with the consumer market in particular becoming more attached to the free and freemium (http://en.wikipedia.org/wiki/Freemium) pricing models. The business model for further API capability became more difficult for several reasons:

1. Network equipment vendors have become adverse to investing in new APIs or evolving existing APIs directly on their network elements. Given their distance from the applications, both in participating in the requirements and taking share of revenue, vendors are reluctant to make investments in capability that enabled the applications without having a near-guaranteed business case predicting the application's success.

2. SDP vendors are faced with the problem of designing to a multitude of existing communications network elements and developing mechanisms to deal with the function disparity in many of those elements. There is also an increase in the number and complexity of web centric APIs and standards being exposed to application developers.

3. Service providers and enterprise administrators are unlikely to make significant investment in broad middleware platforms or incremental investment in evolving legacy, operationally-complex communications infrastructures without a monetisation model that justifies the investment.

**Other Revenue Models**

As a result of the complexity of applying typical monetisation models to the exposure of communications APIs, the technically favourable nature of SDPs, and the dependence of application vendors on the SDP for simplified access to communications capabilities, many vendors have adopted different revenue models.

The model most closely aligned with previous monetisation methods has begun to play out in the Communications Enabled Applications (CEA) industry. Application vendors have begun to vertically integrate their solutions with the necessary SDP capabilities for their solution. By example, IBM leverages their WebSphere Product Family for its application capability and deployment mechanisms and enables WebSphere with capabilities common to SDPs for service creation and interworking with communications networks. Similarly, large IT application companies have built or acquired SDP capabilities to enable them to vertically integrate communications capabilities with application suites. Two significant examples of consolidation to facilitate vertical integration are:

- Oracle acquired BEA, Convergin, and Sun, providing them the ability to integrate the BEA WebLogic SDP, Sun's extensive platform capabilities and Convergin's legacy and next-generation communications network interfaces

- Amadocs acquired long running SDP vendor JNetX as an integration point between their applications and both legacy and next-generation communications networks

By focusing on the vertical integration of applications with the underlying required systems, application vendors are able to monetise the communications capabilities from within their applications. Although they are no longer monetising the APIs directly, the net effect is the same: invocation of communications features results in an invocation of communications APIs from within the integrated infrastructure. Service providers and enterprise administrators are now paying for the value of the API, not the API itself. This permits application vendors to justify new API functionality and incremental API functionality developed on their integrated SDPs as part of the overall application. This approach is especially pragmatic for application vendors when they are re-using the same API for different end-user value propositions. Given the integrated nature of the system, service providers and enterprise administrators are able to focus on the value of the application and the cost of the application as an integrated unit, not as a cost of several disparate capabilities secured from multiple vendors.

This value-based model is still a monetisation of the APIs, but monetisation is not the primary focus. Application providers are able to focus on their core businesses and the APIs become a means-to-an-ends for their value proposition. However, vertical application integration remains a difficult goal for: i) application vendors without the size to acquire or develop their own in-house SDP capability; ii) remaining independent SDP vendors; and iii) network equipment vendors providing their own SDP equivalent offerings or APIs directly from a suite of communications products. Large application vendors wishing to offer vertically integrated capability outside of their core domains of application expertise face the challenge of identifying application opportunities that will lead to successful revenue generation, especially when APIs need to be added or augmented to fulfill the application requirements.

**Coral CEA**

The more substantial number of opportunities that exist outside of vertically integrated solutions is driving a new means of identifying and realising end-user valuable CEAs: CEA developer ecosystems. While providing a developer's community around APIs is not a new concept, several companies and organisations have taken to community focused collaboration around making capabilities available to other application developers.

An example of such a community is the Coral CEA (http://www.coralcea.ca) ecosystem based in Ottawa, Ontario. With founders such as IBM, Nortel, Carleton University, Eclipse, and the Information Technology Association of Canada, Coral CEA offers access to the communication APIs of IBM, Nortel and open source initiatives to members of the ecosystem. Member companies have the opportunity to leverage APIs and expertise in the CEA functional domain so that the member companies can determine the best viable value proposition to end-users. Member companies use the CEA APIs and expertise to augment existing applications or to derive entirely new CEAs. The key value to the ecosystem founders is that they are able to provide existing standards-based capability to member companies to create new value-propositions. The founders may in-turn provide assistance to the members to commercialise new services by channeling the new application/capability to market, joint marketing, or providing a known cost as a service set of capabilities that the member may leverage for commercial sale of their own application. This provides an opportunity to founders or member companies to monetise existing communications APIs; however, it is via the identification and sale of the value-

proposition, not the APIs themselves. Member companies are provided a low-risk opportunity to identify valuable applications and prove them to potential customers without the need of procuring costly CEA capabilities and without the risk of attempting to drive the sale of substantial middleware platforms to their potential customers for applications not yet proven. While there are other examples of such CEA ecosystems, the nature of Coral CEA as a vendor neutral facilitator that provides access to capability based on best fit and low risk development and trialling capability, has permitted it to quickly reach a broad base of companies and establish itself as a reliable keystone in the Ottawa region for CEAs.

**Closing Thoughts**

By examining the continued evolution of the exposure of communications capabilities to applications providers, it is clear that the model of monetising APIs via licensing and transactional based sales can no longer be maintained as the prime means of offering such services. The applications industry has shifted to a value based model, where communications capabilities are a facilitating function, not the defining function. As a result, application vendors, middleware vendors, service providers, enterprise administrators and network equipment providers must continue to define new end-user value propositions, develop and validate them, and bring them to market. By moving to a value based revenue model, and by leveraging vendor neutral business ecosystems, these providers are able to realise revenues more quickly, with more certainty and less risk than by relying on the fading model of building capability and hoping it will be leveraged.

*Craik Pyke is a mobile enthusiast and part-time developer, with a professional background in telecommunications engineering. He has 15 years of experience in the Ottawa high tech industry, working predominantly in communications-centric organisations. Craik is presently a Senior Technical Architect for Carrier Applications and Development Ecosystems at Nortel. Additionally, Craik contributes to the Coral CEA organisation as acting Leader of Technology and Architecture, and is a contributor to iPhoneCTO.com.*

*"There is a way to do it better. Find it."*
Thomas Edison

Companies are always trying to differentiate themselves from the rest of the pack by applying different strategies such as improving customer service, increasing the efficiency of their operations, or reducing their costs. Most of the time, however, these goals are competing against each other for scarce resources, and managers often need to decide to concentrate on one. A small company can effectively and simultaneously accomplish these goals for a fraction of the cost by implementing communications enabled business processes or solutions, which are a set of technology components that add real-time networking functionality to applications. One particular implementation of this framework is the one provided by Coral CEA (http://www.coralcea.ca). Coral CEA is a business ecosystem anchored around communications enabled applications (CEA) functionalities that are offered as building blocks, out-of-the-box components that link the capabilities and intelligence of networks platforms with the power of current applications to provide a new set of features and functionalities.

In this article, we show how a small company called Rezact, located in the ski resort town of Mont-Tremblant, Quebec, successfully implemented CEA capabilities within its own operations using Coral CEA services.

**Business Model**

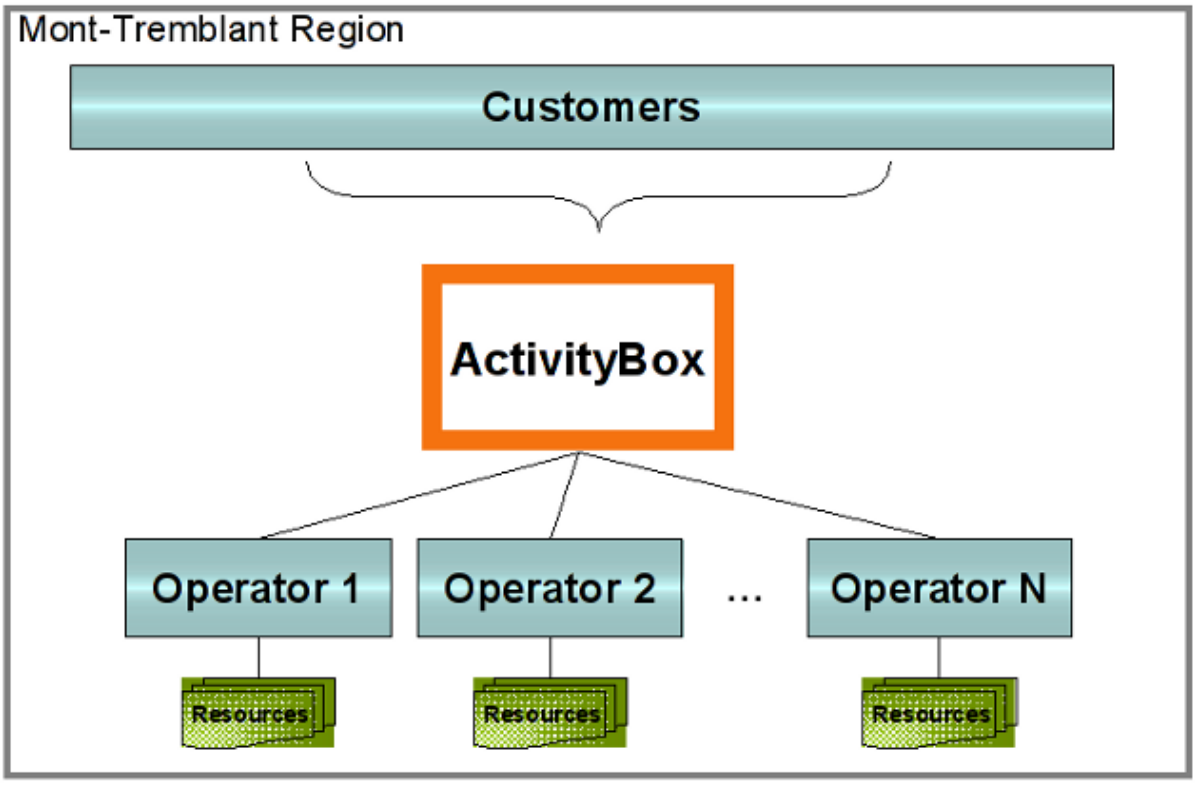
Rezact started its operations in 2006 with the purpose of designing and implementing a new system that could handle the reservation of recreational activities. The business model on which Rezact operates involves three entities. First, there are customers staying at the resort for a brief period of time, usually a week or less. These customers are mostly comprised of families wishing to do something more during their stay in the resort besides skiing. Second, there are small companies (called operators), usually family-operated, that provide customers with a full range of recreational activities as diverse as spa and massage packages, dog sledding, horseback riding, helicopter tours, and rock climbing. The operators are spread throughout a relatively wide geographical area around the resort, and have historically faced the challenge of reaching potential customers and attracting them to their businesses. Third, to attract more customers, operators rely on resellers which sell activities to customers on behalf of the operators while charging them a commission for the service. Under this model, operators that would otherwise struggle to attract customers can reach a lucrative segment for a small commission.

**Activity Box**

To facilitate these interactions, Rezact created Activity Box (http://www.activitybox.ca), an online reservation system that manages reservations for various types of recreational activities like race tickets, horseback riding, spa services or even airplane tours. The system currently serves more than 35 operators managing over 250 activities that are sold by a network of 40 resellers. Since Activity Box was launched in December 2008, it has processed more than 20,000 reservations. The model for the entire system is shown in Figure 1.

Activity Box acts as the connector that links customers staying in a region (in our case, Mont-Tremblant) with a group of operators that provide the activities customers are looking for. Each operator handles through the system its own set of resources to deliver the activities.

**Figure 1: Business Model**




From this perspective, operators rely on Activity Box to offer the best possible service to their customers, improve their operations and increase their revenues. It was under that light that we turned our attention to CEA features to improve our processes.

**How CEA Improved Activity Box**

To improve processes, we searched for limitations or restrictions on the existing operational procedures, analyzed the cause or motivation for those restrictions, and then tried to find ways to reduce or remove them completely. One of those limitations was the way operators get notified of new reservations, which normally occurs by email or fax. Since operators may not check emails or faxes all the time, resellers are forced to create "stop selling periods" a few hours before the start time of the activity to avoid the risk of customers arriving to find out that the operator was not notified or did not have enough time to prepare for the activity.

Blocking reservations, however, can effectively reduce everyone's profits. One way to decrease stop selling periods is to instruct reservation agents to phone the corresponding operators every time a new reservation is created that is close to the activity start time. Since a manual procedure always involves risks, CEA capabilities can streamline this process.

The first benefit obtained from utilizing CEA services was the ability to automatically place a call to the operators to inform them that a new reservation has been created. Once the call has been answered, the operator will hear a pre-recorded message informing them of the new reservation.

Another benefit was an improvement to customer service. For example, sometimes reservations get cancelled due to unforeseen reasons, like poor weather or a broken piece of equipment.

When this happens, the customer needs to be immediately notified so they can decide whether to rebook or to receive a refund for the cancelled activity. Formerly, such notification was a manual procedure solely in the hands of the operator. With CEA services, a pre-recorded message can be automatically sent to the customers affected by the cancelled activity.

The third benefit was an increase of potential revenues to operators. In this case, a customer cancels a reservation and the operator needs to be notified. This is especially important when the cancellation event is triggered by the customer at the last moment in high season periods, when operators are most likely operating at the peak of their capacity. If they are promptly notified of the event, they can react accordingly and allocate the newly freed resource for arriving customers, thus increasing their revenues.

**Implementation**

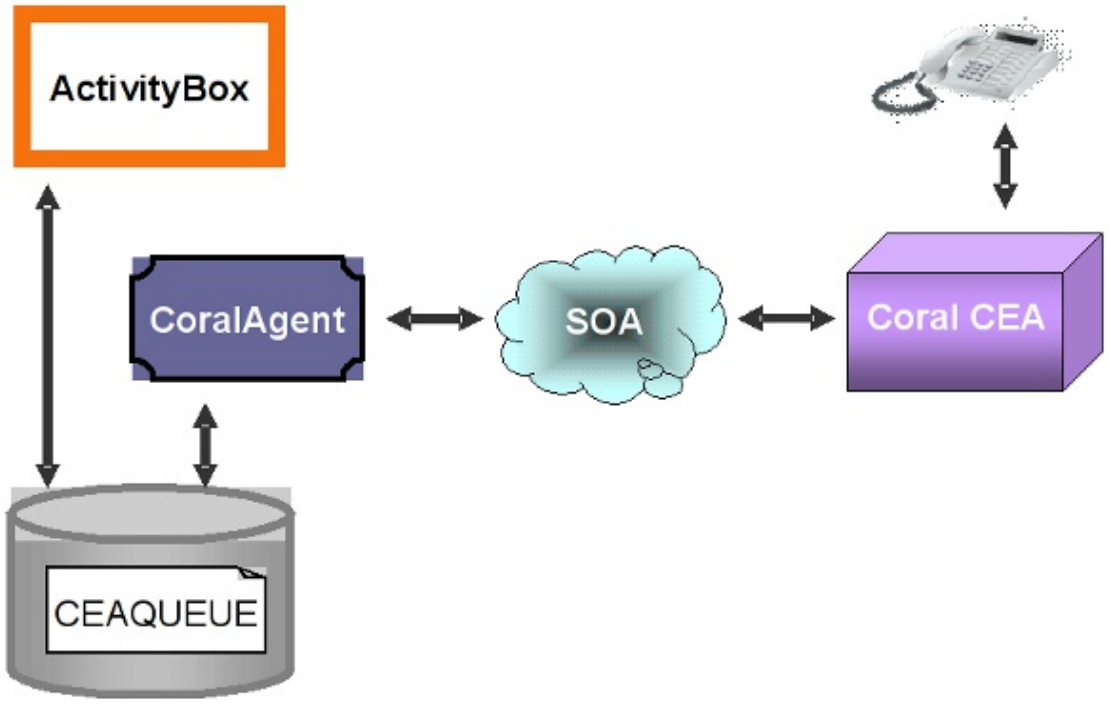
Most CEA implementations, including the one offered by Coral CEA, rely on callable services that use some form of service-oriented architecture (SOA, http://en.wikipedia.org/wiki/Service-oriented_architecture). One of the main features of SOA that needs to be considered when designing a solution is that service calls are usually synchronous, meaning that the client application needs to wait to receive a response from the server. This behaviour could adversely affect our application by creating contention on the Activity Box web server and reducing the future scalability of the system to be implemented within the normal activity reservation process. SOA calls to CEA services need to be decoupled from the web application itself so that performance and user interaction are not affected in any way.

The Coral CEA platform chosen for this project provides several services, including payment gateways and conference capabilities. For our purposes, we were only interested in the communication entry points provided by two application programming interfaces (APIs): i) the Third Party Call Control V3 (TPCv3) API, which allows the creation of communication links between two or more endpoints; and ii) the Audio Call API, which allows an application to play a pre-recorded message to participants on an existing call, as well as to monitor the status of the audio message requested. As expected, these two Coral CEA services only supported synchronous calls.

To solve this issue, we created an agent that served as an intermediary between Activity Box and the Coral CEA server. Every time an event that requires CEA capabilities is detected, Activity Box saves into a common database CEAQueue table the request for an outgoing call, including the name of the audio file associated with the event. The agent, implemented as a Windows service, continuously queries this table to detect any new requests. When a new request is found, the agent submits the request to the APIs and keeps polling the Coral CEA server to obtain an updated status of the request. Each status change is updated back into the CEAQueue table so the client application can be kept informed of the status of any call. The general model of this design is shown in Figure 2.

By decoupling Activity Box from CEA interactions, we eliminate the synchronous problem and provide a safety net to the application in case something goes wrong with the API call. We also reduce the modifications to the client application to just an extra SQL-like instruction to insert the request into the CEAQueue table, keeping the user and business layers free of changes. To provide updated calling status information to the users, Activity Box only needs to query the

**Figure 2: Agent Design**




local table without having to continuously make remote calls to Coral CEA APIs.

With this design we can easily provide CEA functionalities to other applications within our organization as long as they use the shared table. One of the most interesting advantages of centralizing CEA communications is that all interactions with the APIs are completely transparent and developers in the company do not need to know how to make SOA calls, only how to insert a new record in a table.

Some other advantages of this approach include:

- the agent encapsulates the internal mechanisms needed to interact with any CEA provider and exposes them as parameters that can be changed on a configuration file without affecting client applications
- if the requested call did not go through due to technical reasons other than the customer hanging up, the agent can try requesting the call for several times, leaving a reasonable amount of time between attempts. Both the number of attempts as well as the elapsed time between attempts are configurable parameters
- the agent can be configured to place calls between certain periods to prevent the application from calling customers late at night
- call prioritization can be programmed so certain call notifications, such as urgent cancellations, are requested first
- it is possible to programmatically set the maximum number of concurrent requests to be placed to the underlying communication platform. This is useful when the number of physical telephone lines is greater than one, allowing simultaneous calls

- the history of call requests is kept in a centralized location that can be accessed by several applications

Since all the complexity associated with the communication with the CEA service is handled by the agent, the modifications required to Activity Box were reduced to a minimum. First, we needed to modify the database routines that process the creation of a new reservation as well as the cancellation of an existing reservation to insert the required values into the CEAQueue table. The values include the type of event, the name and telephone number of the receiver of the call, the reservation number associated with the call, and its priority. The second modification provides a way for the user to verify how the call went through, by adding visible buttons and links in the application that display the results of those calls, as shown in Figure 3.

**Recommendations**

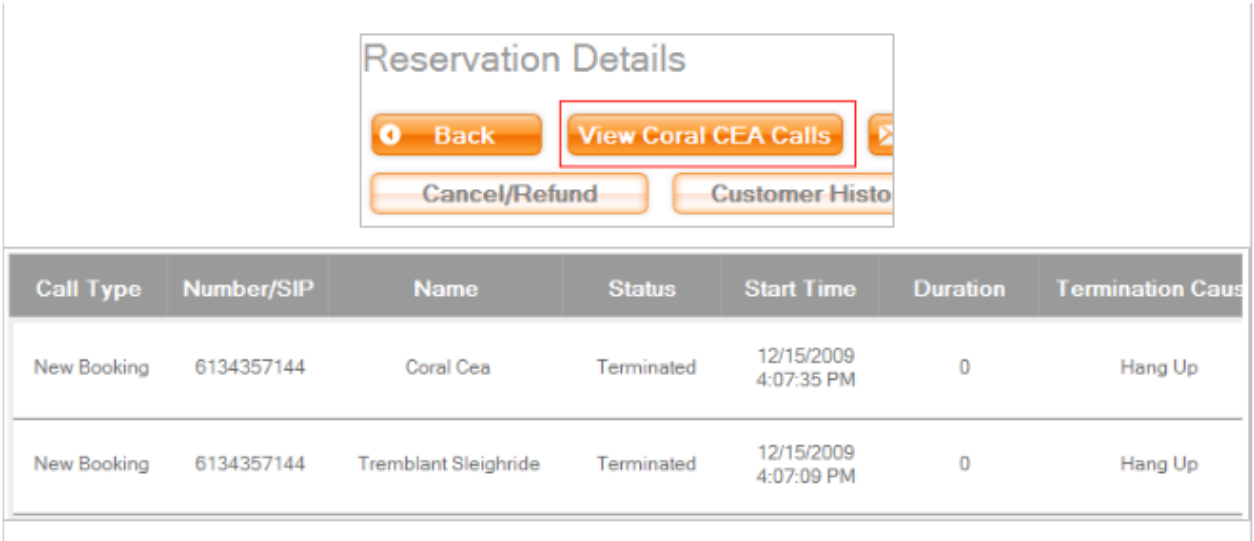
Based on our particular experience implementing CEA features within Activity Box, we can list the following recommendations:

1. Any interaction with CEA services should be treated independently and outside of the regular process or event that triggers the call, to prevent scenarios where there is a limited number of available lines or when the application needs to wait for a response. One way to decouple client applications from CEA services is through the utilization of a Windows service that places call requests and periodically polls the CEA server to update the status of a call or an audio message.

2. When implementing CEA features, in particular those offered by Coral CEA, companies should roughly estimate one month of development time, an estimation that obviously depends on the size of the application.

3. Client applications planning to utilize CEA services should try to forecast, for a given period of time, the expected number of events that could trigger CEA requests. The number of simultaneous calls that can be put through by Coral CEA depends largely on the capacity of the deployed infrastructure and the number of telephone lines assigned.

**Figure 3: Modifications Made to Activity Box**



**Conclusions**

According to our own experience, it is relatively easy for commercial applications to implement CEA features using Coral CEA APIs. A key factor for a successful implementation of CEA services is to keep the client application as isolated as possible from any interaction with the exposed CEA APIs. We have found that once the key elements to communicate with CEA services are in place and a buffer mechanism is used, the remaining effort is solely determined by the interactions between client applications and a common table. This can be done by using the agent proposed here, but other mechanisms could be found according to particular needs and scenarios.

Coral CEA, as a keystone entity and a platform leader offering affordable CEA capabilities, is a diamond in a natural state. It has the key technological elements needed to create a successful business ecosystem, but it requires developers to do some polishing to create ready-to-market applications. It is by leveraging CEA capabilities that a small company can easily improve customer service, increase the efficiency of operations, and reduce costs, just exactly what it needs to differentiate itself from the rest of the pack.

*Daniel Cardenas is a Software Architect at Rezact Inc, a software company dedicated to create a new reservation system called Activity Box. Thanks to the efficiency in the reservation process gained with the new system, our client broke in 2009 a lifetime sales record for a single day of reservations. Before joining Rezact in 2006, Daniel worked for more than 10 years as a Programmer and IT Project Manager for several companies in the finance and manufacturing sectors. Daniel holds a BSc in Systems Engineering from Lima University in Peru and is a recent graduate from Carleton University's Technology Innovation Management Program.*

*"[..] our health and wellness is not simply a responsibility of the state but something we must work toward as individuals, families and communities, and as a nation."*
Roy Romanow

Open source and communications enabled applications (CEAs) are emerging as a vital part of e-Health strategies across Canada. This article provides an overview of lessons learned from our investigation of collaborative telehealth systems delivery in Canadian healthcare. Specifically, the role of open source software (OSS) and CEAs with respect to pan-Canadian patient record management and telehealth service prototyping are discussed.

**Challenges in Healthcare Delivery**

Within the broader blanket of e-Health, which includes health record management, telemedicine has emerged as a unifying power in a largely fragmented system of healthcare delivery in Canada. However, many healthcare service-delivery companies are too small to support the research and development (R&D) requirements of the business on their own, and require partnerships in order to succeed. In combination with mature open source stacks, CEA-based healthcare applications provided by ecosystem partnerships and open collaboration hold promise for needed increases in service interoperability and scalability. Open source component stacks are finding their way into Canada's strategy for creating a cross-country Electronic Health Record (EHR), and for good reason. Some EHR success stories have been publicized in recent years, most notably, the Alberta Netcare system. However, many ambitious, large-scale information technology (IT) infrastructure overhaul projects undertaken by public organizations have been plagued by project cost and scheduling overruns (http://www.canhealth.com/nov09.html#09novstory1).

As a result, many industry experts are calling for a renewed focus on small-scale systems that:

• demonstrate positive results early-on

• start small, but can scale

• make better use of OSS tools

Like many industrialized nations, the accelerated adoption of telemedicine in Canada is fueled by an aging workforce (http://www.wmc-cfb.ca/programs/Aging_Workforce_Final_Report.pdf), demand for equitable and timely access to quality services (http://longwoods.com/product.php?productid=19465&cat=520), and a relative shortage of professional practitioners (http://cdnhomecare.ca/content.php?doc=172). This last factor is particularly noticeable in remote and rural communities in Canada (http://chsrf.ca/final_research/ogc/pdf/macleod_final.pdf). Lengthier wait times to access specialists in the healthcare system and increasing demand for treatment of chronic illness are by-products of these changing conditions.

Home-based telehealth provides the ability to connect scarce specialists with geographically-separated patients. By accommodating patients in their home environment, remote-monitoring systems eliminate a large portion of the travel time required for some home care. Supervising nurses are able to monitor more patients at once, while also increasing overall care by increasing the number of vital checkpoints.

Despite considerable demand for more efficient, customizable, and scalable healthcare solutions, many regional services struggle to find sustainable support models.

The former Canadian Society of Telehealth (http://www.coachorg.com) highlights the lack of consistent reimbursement guidelines as an ongoing barrier to success for telehealth within most provincial jurisdictions (http://cst-sct.org/en/index.php?module=library&VV_DocumentManager_op=downloadFile&VV_File_id=316). Most physicians participate in such services on a per-fee basis, outside of the standard reimbursement process. The agency further recommends that all provinces and territories explicitly reflect telehealth services within fee schedules for physicians. Some provinces, such as Nova Scotia, have managed to institute consistent funding, partly through the help of federal agencies such as Canada Health Infoway (CHI, http://infoway-inforoute.ca), for telehealth expansion and operations using fee-based reimbursement policies and are currently synchronizing their Picture Archiving and Communication System (PACS, http://en.wikipedia.org/wiki/Picture_archiving_and_communication_system) and hospital IT infrastructure systems. On a national scale, however, EHR interoperability remains a challenge. CHI's Blueprint (http://www2.infoway-inforoute.ca/Documents/EHRS-Blueprint-v2-Exec-Overview.pdf) calls for EHR-ready IT and telehealth systems that provide:

- point-of-service access to a shared EHR system, which is itself a collection of synchronized infostructures within provincial domains

- applications that make use of shared EHR data, but cache operational patient data specific to local systems

**Health Record Initiatives**

A major part of telehealth strategies, as well as most anything labeled e-Health, is the EHR. The need to search and share patient profile data is required by every telehealth service in Canada. Typically, these functions are managed through one or more IT systems that access data contained within a patient's EHR profile, depending on the level of regional or provincial coordination involved. In recent years, the idea of developing a pan-Canadian EHR strategy has taken center-stage in discussions about improving equitable access to medical services, a key tenet of Medicare. Canada's vision, largely driven by the federally-funded non-profit CHI, could be described as a system of unified fragments of existing or in-development regional records-management solutions, as shown in Figure 1.

A tie-in for the standards-first message promoted by the CHI is the EHRS Blueprint. The Blueprint depicts user storyboards which are diagrams that map a patient's interaction with the medical community throughout the continuum of care across professional domains. Figure 2 provides a storyboard representing a patient's first visit within the chronic obstructive pulmonary disease treatment domain.

From an implementation perspective, the CHI does not stipulate a great deal of technical requirements for its paid service provider members. Instead, it provides access to licensed standards material such as HL7 (http://hl7.org) and SNOMED CT (http://ihtsdo.org/snomed-ct). By providing guidelines and best-practice documentation, CHI hopes to build an interoperable patchwork of infostructures across the country that professionals can tap into. The push to use a messaging standard like HL7 is not a revolutionary concept, as most EHR solution providers support it, but it does encourage interface efficiency when designing integrated telehealth services.

**Figure 1: CHI's Infostructure Vision**

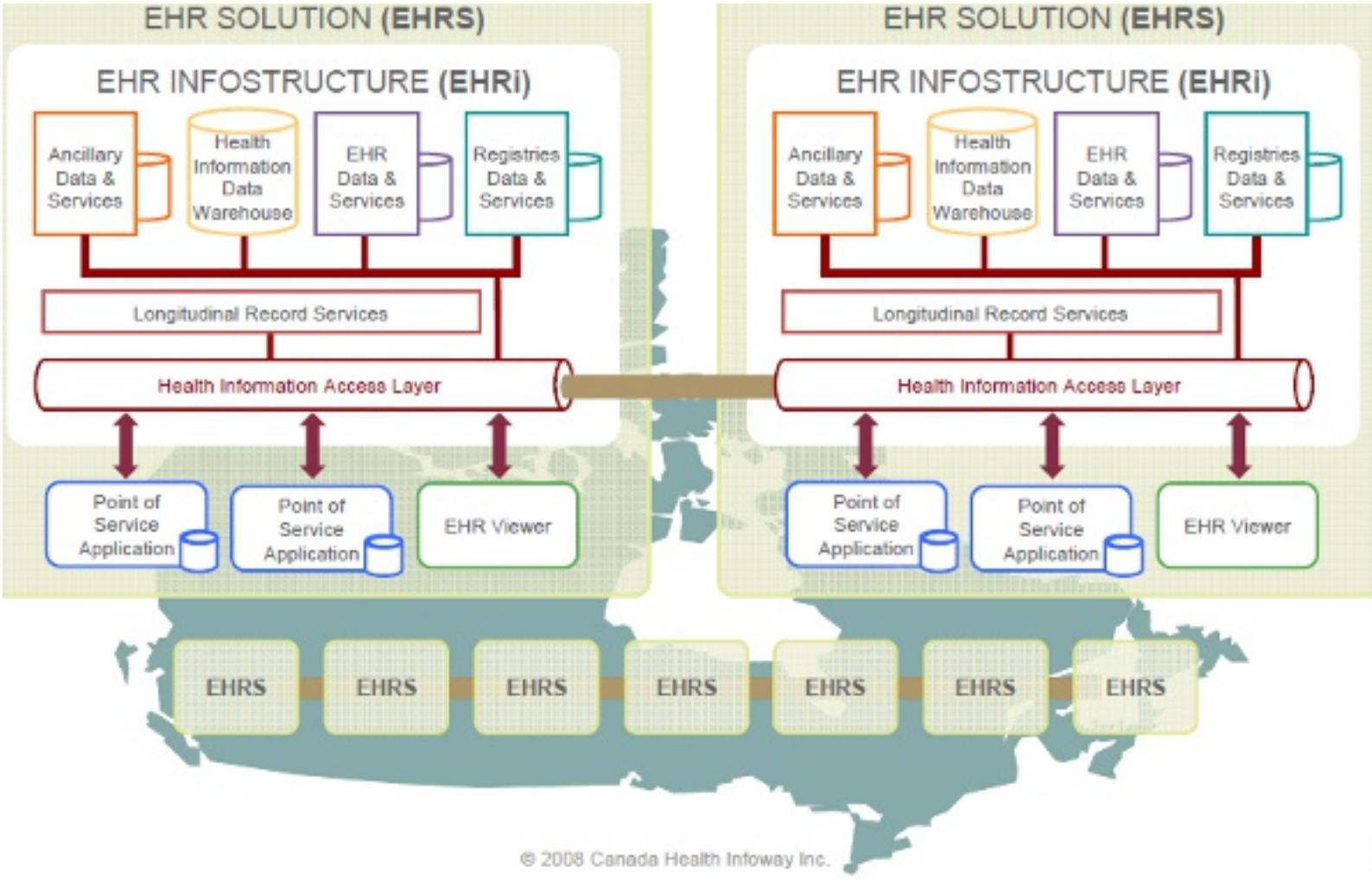
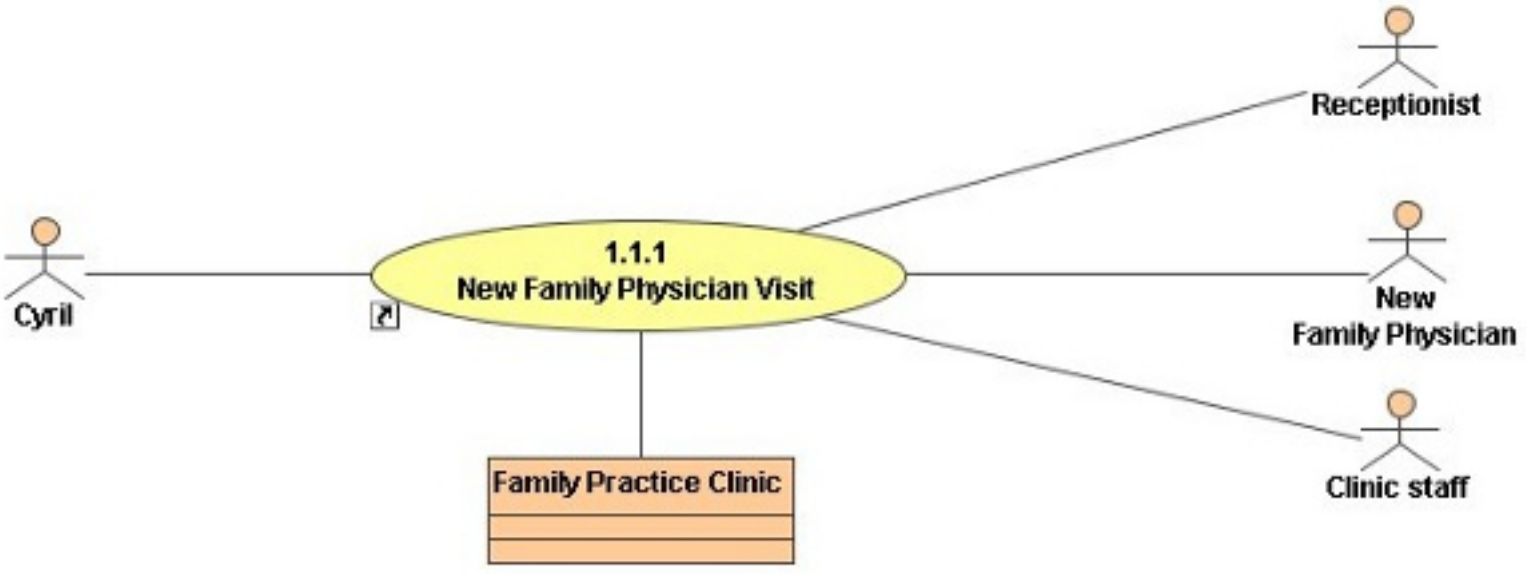



**Figure 2: Storyboard Representing First Patient Visit**



The Infoway Reference Implementation Suite (IRIS, http://secure.cihi.ca/cihiweb/en/downloads/HL7Can_IrisUpdate.pdf) is arguably the most important initiative to emerge from CHI's game plan, from an entrepreneur's perspective. IRIS, released under an academic open source license, ratifies the HL7 protocol stack by leveraging a veritable who's who of open source components. Providing a layer of abstraction between application and data sources with IRIS might be the right ticket for many small entrepreneurs to contribute a compatible piece to an otherwise complex and distributed infrastructure. Learning from Alberta and Nova Scotia's experiences in expanding their regional IT systems to support a common EHR, CHI's approach to support both large and small players to drive the next-generation of user-centric, innovative telehealth solutions is forward-thinking. By not restricting providers to a single, all-encompassing format, they are opening the door to more efficient ways to manage and share patient data regionally and across Canada. A quick scan of "open source EHR" projects on Wikipedia (http://en.wikipedia.org/wiki/List_of_open_source_healthcare_software) reveals a list of at least 28 separate initiatives to create a standard record format for various healthcare domains. Even some early hierarchical database pioneering projects, such as the over-40-years-old MUMPS (http://en.wikipedia.org/wiki/MUMPS), are finding new life in today's collaborative environments.

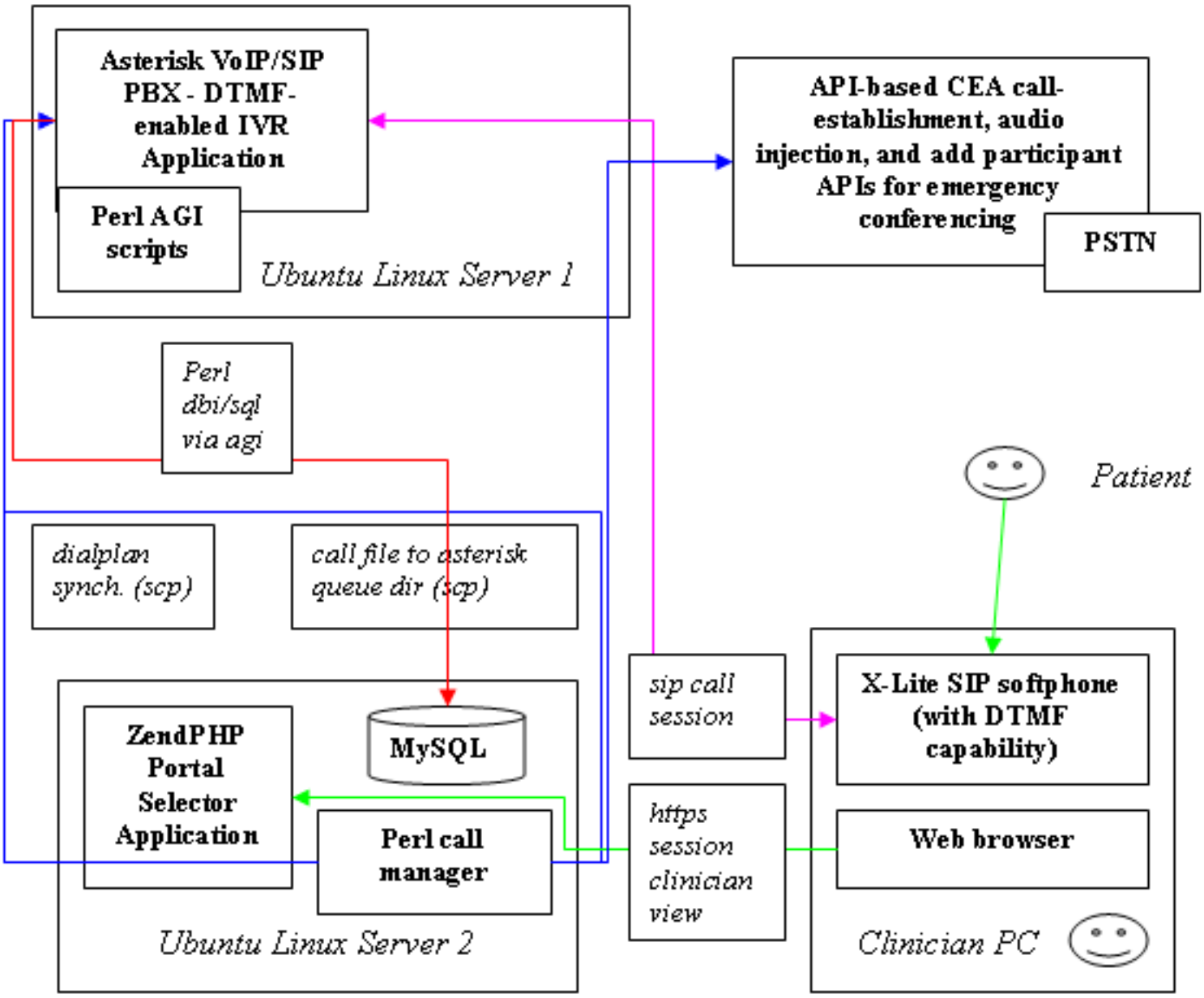
**Prototyping Telehealth Innovation with Open Source and CEA APIs**

Beyond EHR, community-driven telehealth initiatives anchored on other open source components and CEAs may address some of the long-tail requirements and funding limitations felt by regional operators. In the context of telehealth, prototyping is manifested through process and protocol testing in clinical trials. Considering the legacy of top-down, large-scale projects in healthcare, such as videoconferencing infrastructure deployment, application prototyping is likely to increase in popularity as providers focus on customized solutions for segment markets at the end of the long tail and bottom-up scaling of solutions. Building telehealth services with open source software components invites the use of prototyping through low-cost experimentation of design. One such prototype telehealth application, is outlined in Figure 3. In this case, the simple application could be used to setup and schedule call-outs of patient-centric questionnaires that aim to reduce the likelihood of early hospital re-admission due to a patient's non-compliance with their medication. In the described prototype, a service provider could leverage the open source asterisk PBX (http://en.wikipedia.org/wiki/Pbx) and call-session management functions through open application programming interfaces (APIs) supported by third-party providers such as Coral CEA (http://coralcea.ca) or Twilio (http://twilio.com). Through a web application interface, a clinician could setup questionnaires and track a patient's response history in order to enhance the spectrum of care. One can imagine a tie-in with IRIS in order to share and coordinate this patient information with other relevant clinical applications.

**Testing the Concept**

As a follow-up experiment in early 2010, the questionnaire-based interactive voice response application prototype concept was built and tested as an academic exercise within an IRAP-funded (http://www.nrc-cnrc.gc.ca/eng/ibp/irap.html) project to help entrepreneurs explore the value of CEAs and the Coral CEA sandbox. The result of this initiative was a demonstration web portal that tied together Coral CEA call-conference management web services and an isolated

**Figure 3: Clinical Questionnaire Engine Prototype Design**




asterisk system which was able to interact with a locally-registered X-Lite SIP soft-phone in order to setup, place, and record simulated medication-compliance questionnaire calls. The system used pre-recorded text-to-speech voicefiles (to simulate a dynamic engine) linked with questions, together with clinician-supplied call-flow logic, that formed a typical call-out patient compliance questionnaire.

Using database-registered input and action codes associated with questionnaire voicefiles, the system was able to determine the sequence of questions to play for the patient during a call, based on key digit responses received.

The Coral CEA sandbox services provided the ability to create a simulated emergency conference call between a patient and emergency attendant staff using a secondary tie-in application that monitored call state and question responses. In the event of an emergency condition, the patient was asked to hang-up the phone and to expect an immediate call from the nursing staff. The system was then tasked with:

1. Acquiring a call session with the monitoring staff.

2. Providing an automated notification of the situation once a connection was made.

3. Attempting a third-party conference-call with the patient.

Figures 4 through 6 provide a high-level visual representation of the prototype's clinician web interfaces and general functionality. The prototype's logo is licensed under a Creative Commons Attribution 3.0 License. Figure 4 shows the initial web portal screen following authenticated login by the clinician. The screen to the right is a capture of the patient setup tab, within the web portal's clinician management area. In this space, the clinician can register patients within the application, and find or modify existing patient profiles. Figure 5 depicts the clinician's questionnaire setup section of the prototype's web portal. Here, the clinician can sample pre-loaded questions in the application's database, organized by category and function type (e.g. generic or relevant to a particular condition or medication). Questions can be added to a questionnaire editing area, where question ordering and response logic can be modified.

Once questionnaires have been added to the system, the clinician is able to view general logic and call flow via a separate dashboard area. In Figure 6, a sequence of callout questionnaire events is shown:

1. A clinician schedules a questionnaire call event for a given patient.

2. The system confirms the scheduling request.

3. According to the start time, frequency, and duration parameters provided, the call manager organizes call queue events within the asterisk system. A tested call is shown in the last frame as tested using a registered X-Lite SIP softphone client.

**Figure 4: Demonstration Prototype Post-Login and Patient Setup**

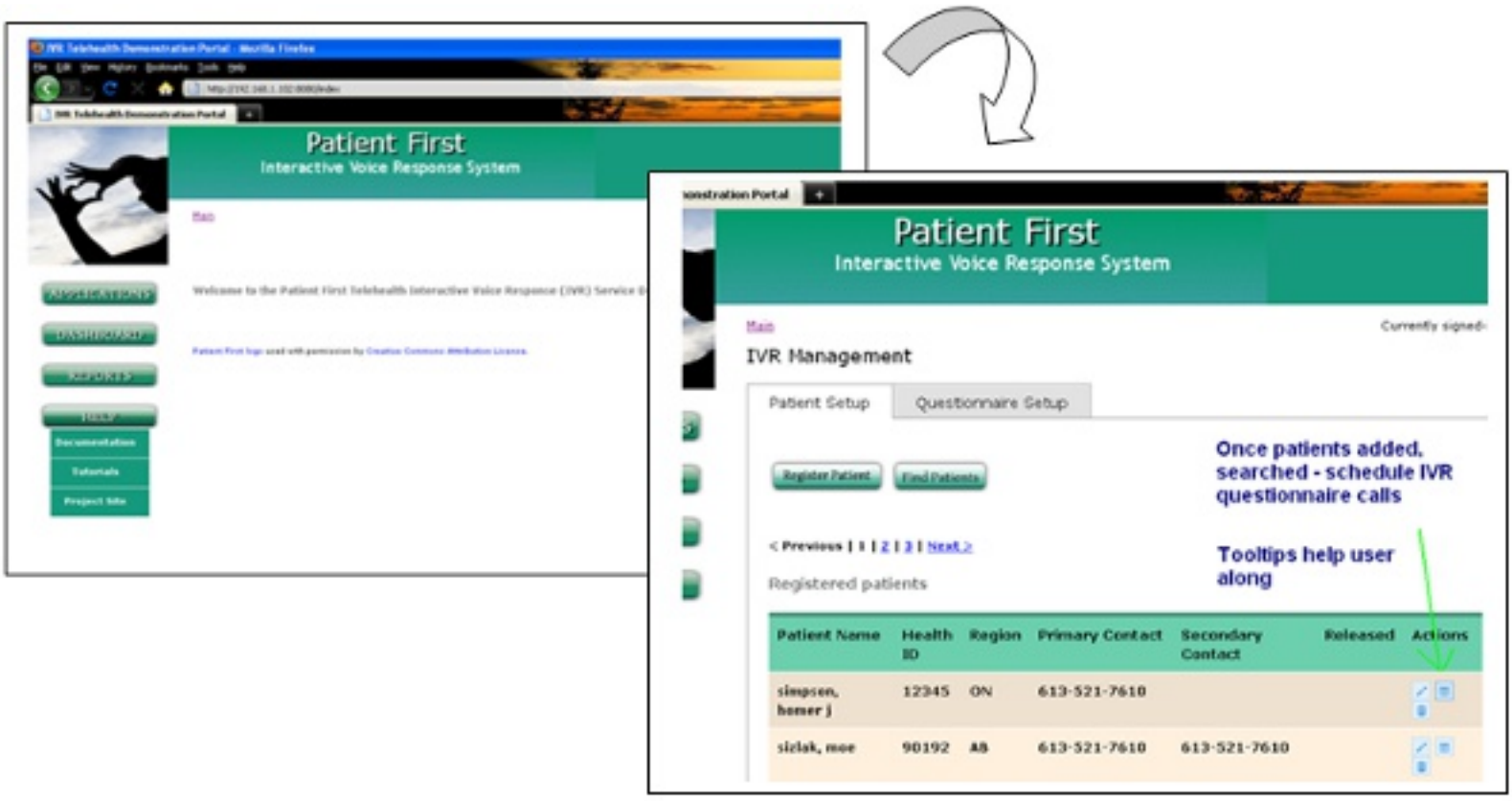


**Figure 5: Questionnaire Setup Functions**

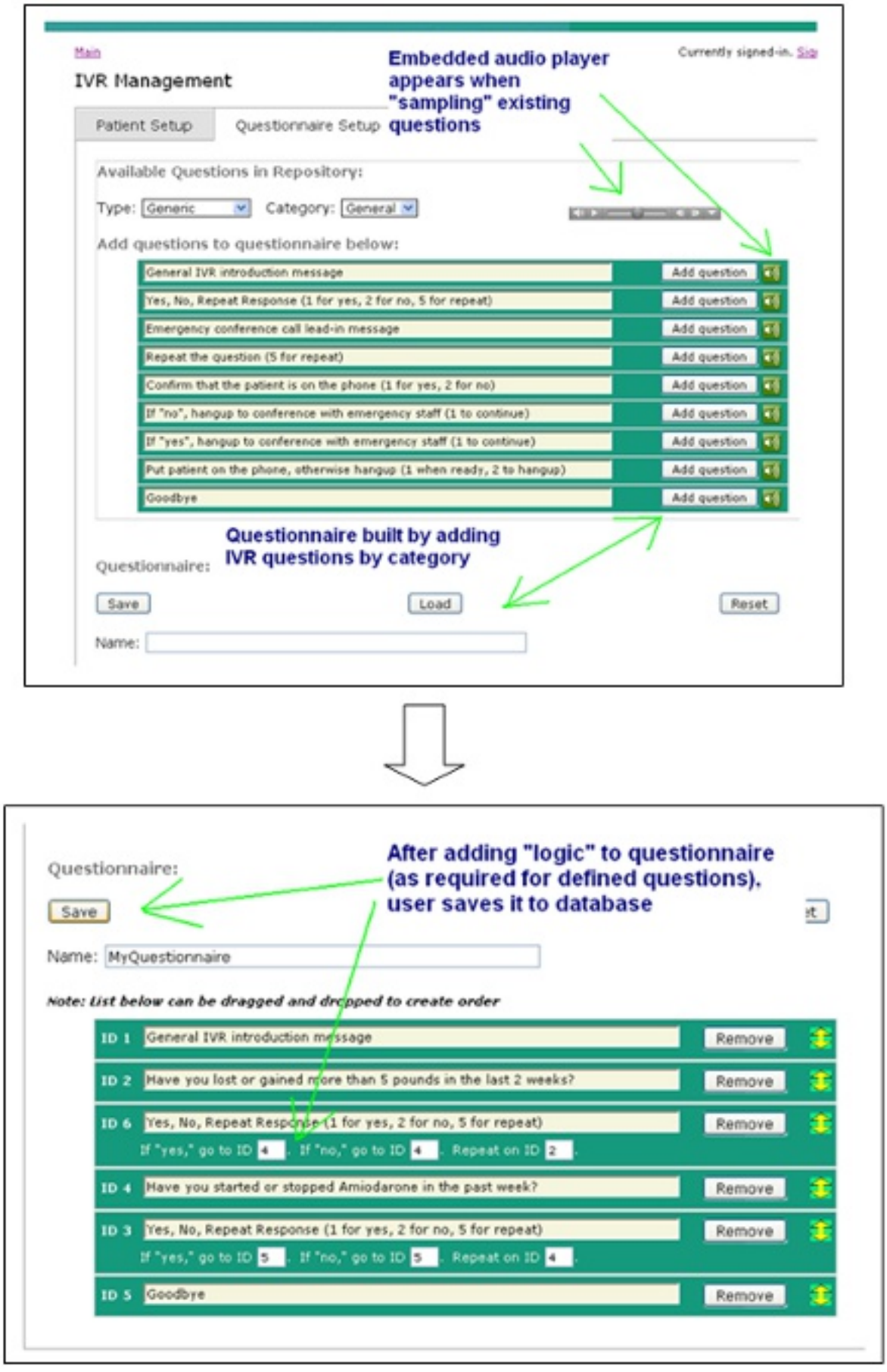
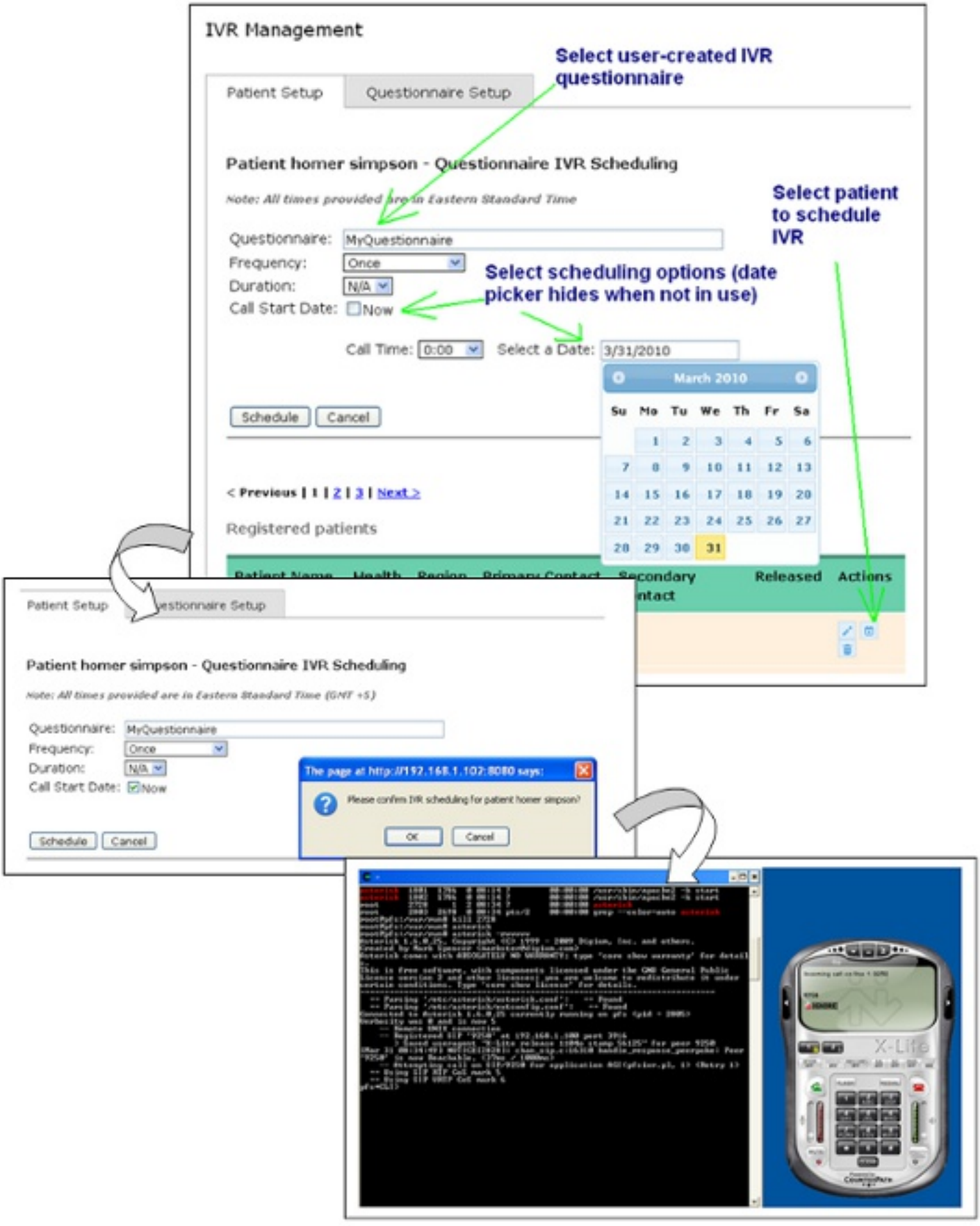


**Figure 6: Test Call Setup and Execution**



**Conclusion**

The application of collaborative frameworks in the healthcare domain provides an opportunity for innovation and leadership in an arena traditionally dominated by large-scale IT initiatives. Increasing pressure by federal and provincial governments to implement a common EHR system across Canada and mature open source driven telecommunications stacks have provided the foundation for a collaborative revolution in healthcare delivery. Entrepreneurs seeking to innovate in this space should heed the message of collaboration and partnership, leveraging commoditized data-management services from provincial EHR infostructures. Similarly, innovations in community-led CEA infrastructure development should play a pivotal role in expanding shared provincial communications infrastructure capacity and standards to support periphery application innovation.

**Recommended Resources**

Canada Health Infoway EHR
http://knowledge.infoway-inforoute.ca/EHRSRA/index.html

CHIRIS project on Sourceforge
http://sourceforge.net/projects/crrs

*Andrew Ceponkus graduated from the University of Toronto's Engineering Science program in 2001, after which he spent nearly 10 years working in various software and networking technical roles with companies in Ottawa and Toronto. He now works as a team lead and project manager within IBM's Tivoli Provisioning Manager Quality Assurance team, and expects to graduate from Carleton University's Technology Innovation Management program this term.*

*"He who receives an idea from me, receives instruction himself without lessening mine; as he who lights his taper at mine, receives light without darkening me. That ideas should freely spread from one to another over the globe, for the moral and mutual instruction of man, and improvement of his condition, seems to have been peculiarly and benevolently designed by nature, when she made them, like fire, expansible over all space, without lessening their density in any point, and like the air in which we breathe, move, and have our physical being, incapable of confinement or exclusive appropriation. Inventions then cannot, in nature, be a subject of property. Society may give an exclusive right to the profits arising from them, as an encouragement to men to pursue ideas which may produce utility, but this may or may not be done, according to the will and convenience of the society, without claim or complaint from anybody".*

Thomas Jefferson

A platform of CEA building blocks, such as the out-of-the-box capabilities of the Coral CEA Sandbox (http://coralcea.ca/content/coral-cea-sandbox), provides companies with the capability to quickly build new innovative products and services. Key considerations for users of the sandbox include intellectual property (IP), licensing, and any other dependencies inherited from use of the sandbox assets. This article presents some background on this topic and examples of how to address the associated implications.

**A Little Background with a Software Bias**

Intellectual property rights (IPR) refer to the exclusive rights granted to the creators of original works. By general convention, IP is comprised of products of the "human intellect that have commercial value and that receive legal protection" (http://www.nolo.com/products/patent-copyright-&-trademark-PCTM.html), encompassing "creative works, products, processes, imagery, inventions, and services" under the protection of patent, copyright, trademark and trade secret law. Focusing narrowly on IPR as it pertains to software, we can expand on the concepts of patents and copyright:

**Patentability:** software patents typically fall in the domain of utility patents, where they are captured under the description of a process. They became prevalent in the US in the 1980's but were typically associated with software that interacted with hardware and related devices. Software patents are also applicable from a Canadian perspective, conditional that the software is integrated with a technology that is traditionally patentable.

**Copyright:** for purposes of the copyright law in general, software, including object code which can only be read by a machine, is typically considered a literary work. A software copyright owner has the exclusive right to: i) reproduce the work; ii) create derivative works; iii) distribute copies of the work; and iv) publicly display the work. Computer programs are protected as literary works within the meaning of Article 2 of the Berne Convention (http://wipo.int/treaties/en/ip/berne/trtdocs_wo001.html#P85_10661) and such protection applies to computer programs, whatever the mode or form of their expression.

Although there may be different definitions by jurisdiction, all forms of software are protected by copyright. And when it comes to the application of an IPR within the software domain, the leaning tendency is towards a copyright directive rather than patents.

**Software Licensing**

What is the relationship of copyright to license? In plain terms the distinction can

be made as follows. If we make the analogy of code as a home, copyright can be said to be the ownership deeds of the home, and unless you assign those deeds to another entity, you retain ownership of the home. Licensing isn't about giving away that ownership, it is about setting the rules by which the home owner allows others to use their home.

Andreas Constantinou proposes that the use models and adoptions of specific licenses in different software domains are dependent on the needs and directions of the perspective ecosystems, and also the mechanisms that are provisioned to cater to member use patterns (http://osbr.ca/ojs/index.php/osbr/article/view/1049/1008). When we consider software and its associated applicability of copyright, we typically think in relation to traditional client/server software rather than the newer software-as-a-service paradigm. Our concept of software needs to be updated. We need to address the questions of whether and how the contemporary software components and services which make up CEA should be licensed and whether the traditional software licenses can be applied. The question of whether "licenses are a legal artefact applicable to services" as propositioned by Gangadharan & D'Andrea (http://jiclt.com/index.php/jiclt/article/view/66/65), has been asserted positively in the previous work of Gangadharan (http://static.digns.com/uploads/doctoral_school/documents/phd-thesis/XX/gr_gangadharan.pdf).

We need to highlight the differences between the contemporary and traditional components, to identify applicable license criteria. Web services "are not targeted as standalone applications" (http://www.computer.org/portal/web/csdl/doi/10.1109/AICT-ICIW.2006.124) and, unlike traditional software, "web services do not execute over any specific hardware or software platform".

Moreover, when we compare the differences in the make up of web services, we can see that these differences center around the concepts of: i) hosted environments; ii) reuse models; iii) composition models; and iv) data (http://disi.unitn.it/~gr/PLWS.pdf).

**Implications**

As with traditional software, a web service can be proprietary or open. Gangadharan proposed a means of capturing the licensing patterns of web services, as summarized in Figure 1.

With this representation comes the implications for platform providers within the CEA domain. CEAs are building blocks that can be leveraged, reused, and combined. These components can be many, and their derivative web service complementors can have followed any of the patterns defined. Therefore, it is essential that CEA platforms provide mechanisms or incorporate process hooks that allow the user/member communities to have visibility of such dependencies, or a means by which the steps to address and resolve any associated incompatibilities can be automated.

**Thoughts for Resolution**

The assets deployed within CEA based platforms comprise various definitions, from the underlying building blocks, to the publicly visible enabler functionality of the web service components. Two suggested means by which the IP nuances within CEA could be addressed are through:

1. The utilization of various software IP audit services. The key players in this space are Black Duck Software (http://blackducksoftware.com/services/professional-services/assessment), Palamida (http://palamida.com) and Ottawa based Protecode (http://www.protecode.com).

**Figure 1: Web Service License Patterns**

| Web Service Type | Licensing Terms | Options | License Examples |
|---|---|---|---|
| Proprietary | Separate licensing terms | | • Amazon Web Services Licensing Agreement<br>• Yahoo APT API Licensing Agreement |
| | Limited execution | • Limit requests<br>• Limit results<br>• Limit quality | |
| Open | Release on same terms on which it was based | | • AGPLv3 |

The majority of these services primarily focus on the compatibilities of open source licensing.

2. Employing or incorporating into the CEA governance platform a machine readable and automatible syntax for capturing the IP assets. A possible mechanism would be a solution based around Rights Expression Languages (RELs, http://osbr.ca/ojs/index.php/osbr/article/view/465/413).

An example syntax that specifically focuses on automation prospects relates to ODRL-S (http://dit.unitn.it/~gr/ODRLS,pdf) a profile which is based on the Open Digital Rights Language (ODRL, http://odrl.net/1.1/ODRL-11.pdf). It is provided as a means to express a service license so that any services can automatically interpret the licensing dependencies from the clauses it presents. The five applicable clauses are: i) subject; ii) scope of rights; iii) financial terms; iv) warranties, indemnities, and limitation of liabilities; and 5) evolution.

A key benefit of utilizing a methodology such as ODRL-S is that service level agreements (SLAs) requiring negotiations between a service consumer and provider could be circumvented. The license can now take the form of unilateral statement, specified by the provider to one or more consumers, without involving protracted negotiations for each engagement.

**Conclusion**

We expect that CEA will provide numerous new solutions and business prospects for many years to come. In order to ensure an uninhibited user community and open innovation, the providers of CEA platforms need to address the underlying IP needs of platform users. By addressing these needs and provisioning mechanisms for IP clarity, the organization removes impediments to productivity. Such mechanisms currently exist and the potential and advantages for automating these processes are evident, based around the many and varied interactions that need to be supported.

*Patrick O'Halloran is a graduate of the Computer Engineering Program from University of Limerick, Ireland. He is currently studying Technology Innovation Management from Carleton University's Department of Systems and Computer Engineering. Mr. O'Halloran is a Staff Design Engineer with Xilinx Inc. (http://www.xilinx.com), within the Xilinx Design Services group, and has been working on varied consultancy projects in this role for the past 10 years. These projects have centered around Xilinx's FPGA technologies and their application to many industry verticals. He has varied interests which range from Technology Innovation, Real-Time systems and IPR in the technology domain.*

**Recommended Resources**

Software IP Management Blog
http://www.squidoo.com/software_ip_management

Software Licensing in the Cloud
http://www.elastra.com/sites/default/files/elastra_licensing_in_the_cloud_web.pdf

"*Communication is the real work of leadership.*"

Nitin Nohria,
Harvard Business School

For technology companies seeking to harness powerful open source technologies, few can argue against the usefulness of modern Web 2.0 platforms. Considering how ubiquitous the Internet has become, an entrepreneur would certainly be ill-advised to not use some Web 2.0 platform to facilitate access to Communication Enabled Applications (CEA).

This article enumerates the lessons learned by one startup in order to demonstrate the need for a balanced approach to CEA to facilitate access to untapped markets.

**Preparing for CEA**

The overall premise of CEA is simple but comes with one important requirement: company-wide data coherence. CEA technology cannot possibly bear fruit in a company environment where the data is a mess. Examples of non-coherence include dozens of spreadsheets emailed around, printed forms manually entered by staff, and databases that occasionally talk to one another via import and exports. Such painful problems must be addressed before a company can consider embarking in the CEA space.

Once information is coherent and the company has centralized its data in a robust and secure database, CEA benefits become possible. By adding links to employees, suppliers and partners into the database, the company can become leaner and more responsive as data duplication is eliminated between parties. The promise of the paperless office begins to bear fruit with company contacts, and the firm finally becomes a candidate for real scalability and global success.

Rich Internet Application frameworks such as SmartClient (http://www.smartclient.com) or GWT (http://code.google.com/webtoolkit) can be used to implement a secure communication link with external contacts using highly interactive web pages, enabling access to company contacts and actual customers. Now that simple web links can point to highly interactive applications, a site that provides a quality interactive experience is much more adept at converting the passerby into a potential customer.

Customer buying patterns are pointing toward an expansion in web-based transactions, versus the traditional brick and mortar model. Most companies would curtail their own growth by not leveraging web-based CEA technology to reach the new generations of customers, whose first step when needing a product or service involves a time-saving scan with a search engine.

Once a good Web 2.0 platform has been integrated into a company's system, the benefits of the company's coherent data systems to reach even more people can be extended with the telephone. Why invest in this older technology? Compared with the interactivity and delivery bandwidth of a quality Web 2.0 interface, a voice interface is a limited interaction method and the restrictions inherent in the medium can be frustrating. Nonetheless, the telephone is a trusted and well understood tool, and its inclusion in your CEA portfolio can affect how your offering is perceived by the mainstream consumer.

**Benefits of Integrating Telephony**

Our startup company felt its marketing message left many people indifferent to our offering. Seeking to improve the spontaneous appeal of our publicity message, we created a rough telephony demo and proceeded to observe the reactions to our new marketing material.

We were profoundly surprised to find that phone features are:

1. **In high demand:** potential customers were much more excited at a rough demo of simple phone interactivity features than many of our Web 2.0 features.

2. **Reassuring:** phone access is available and reliable while the Internet is not. For mission critical companies, forcing your customers to use the web means gaps in your service to them.

3. **Scalable:** be it an automated 1-800 number or an automated incident call system, modern telephony ecosystems can deliver features that can free your staff from time-wasting calls.

4. **Easy to sell:** entrepreneurs should note that many potential customers have difficulties distilling Web 2.0 features to tangible benefits in their lives. Telephone features are easy for customers to understand and pay for.

The bottom line of our investigation: a rough demo of phone features gathered more attention from potential customers than the snazzy web platform we had invested years constructing. Because our new pitch now involved a tool that was readily understood, the perceived value was much higher and our marketing collateral became more concise and effective.

This grounding effect to the mainstream customer was so significant that our new phone features take a commanding portion of the pitch we give potential investors. Because of these features, we're now perceived as being able to finally reach the mainstream, not just the early adopters.

If your company finds itself with lukewarm market interest, consider adding

some phone features to observe if mainstream customers are able to relate to your offering in a more intimate manner. With so many products and services offered by the global economy, any feature of your offering that utilizes this trusted old tool makes it easier for potential customers to map features to benefits, and may become a powerful differentiator in your target market.

Be it a dial-in 1-800 number providing customers key information or automated calls delivering key business events, phone features can in many cases bring extra value to a CEA portfolio and further help distinguish an organization from its competitors.

**Telephony Obstacles**

Our company incurred many setbacks as it discovered that entering the brave old world of telephony is difficult. There is a world of difference between making test calls on a quickly installed asterisk (http://www.asterisk.org) system and developing a robust and scalable telephony platform that will integrate into your corporate CEA infrastructure. Invest your time in the wrong part of the asterisk technology tree and you can pay dearly later on attempting to increase the robustness and scalability of your system.

While asterisk dominates the open source telephony world, the phenomenal business success of Digium (http://www.digium.com) has brought about a plethora of partner companies offering their wares to the asterisk ecosystem. It can be difficult at first to separate the obsolescent from the leading-edge, and getting a sense on how to reach best practices is nearly impossible without someone who understands the field.

Here are seven lessons we have learned over the years:

1. **The asterisk ecosystem is vast:** take time to get oriented with the many companies and groups that make the ecosystem powerful and dynamic. As you encounter an unrecognized technology, take the time to research and get oriented about its relationship to the ecosystem.

2. **Start with a good footing:** few users build asterisk boxes from scratch. Trixbox (http://www.trixbox.org), PBX In a Flash (http://pbxinaflash.net) and Digium's AsteriskNOW (http://asterisk.org/asterisknow) are the top Asterisk distributions and each is well worth your study and consideration.

3. **Find a supportive community:** a community of enthusiasts can greatly ease adoption pains. Locate forums early on and learn from the technical savvy of its members.

4. **Some parts of the asterisk technology tree are antiquated:** for instance, the dialplan asterisk programming language is a poor development platform to build robust bridges to CEA systems. AGI (http://www.voip-info.org/wiki/view/Asterisk+AGI), FastAGI (http://www.voip-info.org/wiki/view/Asterisk+FastAGI) and AMI (http://the-asterisk-book.com/unstable/asterisk-manager-api.html) are more robust platforms to programmatically control asterisk using your favorite language and development environment.

5. **Enlist the help of someone who's done this before:** a few billable hours at the beginning of your evaluation can mean thousands of dollars saved later on, especially if you adopted the wrong approach to solve early needs. Enlist an expert who will listen to your needs and steer you toward the best techniques.

6. **Ramp up with an expert by your side:** it is one thing to get a good asterisk-to-CEA box handling a few dozen calls and another thing entirely to scale up to handle the world. Scalability issues are understood in the technical forums and some expert advice at the beginning can steer you toward the right tools and hint at the most cost-effective services needed to host your telephony servers. A not-for-profit CEA ecosystem such as Coral CEA (http://coralcea.ca) can be instrumental in this regard.

7. **Experiment and have fun:** you now have the power to surpass systems costing tens of thousands of dollar. Gain energy, enjoyment and confidence by focusing on fun ways to ramp up your mastery of asterisk. For example, try some of Nerd Vittles (http://nerdvittles.com) ideas while you evaluate what the technology can do. Perhaps some of these techniques can provide value to your customers.

**Closing Thoughts**

While the full promise of open source CEA development is only available to companies that have database coherency, the benefits of CEA are too significant for any company to ignore. Operating a company where the data is a mess is not only crippling to scalability and growth, it also prevents the best elements of CEA technology from becoming possible.

By providing many of the tools for maximizing the benefits of CEA, be it the integration of telephony features with asterisk, the re-structuring of company documents with the Alfresco (http://www.alfresco.com) content management system, or the integration of various data sources with the MySQL open source database, open source technology continues to deliver an unbeatable value proposition. No company should ignore the remarkable benefit to cost ratio that open source solutions can bring to the CEA space.

*Jean-Pierre Poulin is an entrepreneur currently evolving a high-tech startup in the Ottawa area. Sharing his experiences ramping up CEA skills, Jean-Pierre provides a web-based consultation service to companies seeking to orient themselves on their options before investing in expensive development.*

*"Electric communication will never be a substitute for the face."*
Charles Dickens

Customer relationship management (CRM, http://en.wikipedia.org/wiki/Customer_relationship_management) software is used to manage and enhance a company's interactions with its customers. Typically, CRM software integrates well with other communication software, such as email, but includes little or no integration with telephone or video conferencing systems. For companies that interact frequently with their customers or internal teams by telephone or video, this greatly limits the usefulness of their CRM system. Ideally, a CRM system offers flexibility to allow communication with customers in a variety of ways and provides consistent reporting and logging of these interactions regardless of which form of communication was used. This allows for greater insight from interactions with customers and helps better understand how to meet customer needs.

This article describes our recent experiences as we set out to integrate communications services provided by Coral CEA (http://www.coralcea.ca) into an open-source CRM system. Coral CEA is a platform that provides developers open application programming interfaces (APIs) to easily integrate powerful communication features into web applications. In our case, we were interested in enabling telephone, telephone conferencing, and video conferencing services within a CRM application.

However, the article is not just relevant to CRM users, since it illustrates how powerful communication services can be easily added to almost any existing web application. After reading this article, you will have a better understanding of the basics of CRM, how better communication improves the experience for both the company and the customer, and you will hopefully be encouraged to consider integrating powerful communication services into your own web applications.

## Extending a CRM System

CRM software is typically used for marketing, sales, customer support and technical support. Within each of these uses, several mediums of communication are possible between the CRM user and the customer, including telephone and email. Most CRM systems have tight email integration, where email addresses are clickable links that automatically launch an internal email client. However, the telephone system is not as tightly integrated. In some cases, the company's PBX (http://en.wikipedia.org/wiki/Pbx) system is connected to the CRM so that calls can be recorded or incoming calls can display the customer information automatically.

Our goal was to create a communications enabled CRM with telephone integration that met or exceeded the standard level of email integration. To accomplish this goal, we needed to:

- make calling customers as easy as clicking on a phone number
- record all conversations with customers so that they can be listened to at a later date
- transcribe all conversations to text using speech recognition
- enable conference calling

However, there is more to communication than voice features. We decided to extend the communications capabilities of the CRM even further by integrating text, audio and video. For this, we took advantage of the open source web conferencing project BigBlueButton (http://www.bigbluebutton.org).

BigBlueButton offers the following features for integration into the CRM:

- text, audio and video communication over the web between the CRM user and customers
- access from a web browser so that any machine can access the CRM
- desktop sharing so that the customer or the company can share the view of their computer
- slide presentations so that the CRM user can present to customers

Altogether, extending the CRM using these communications features gives us the following benefits:

- reduced communication costs (by not having to pay for or manage PBX systems) and video communication servers
- greater diversity of communications options for users
- increased information and metrics captured within the CRM
- superior communication experience

**Selecting a CRM**

There are many different open source and commercial CRM applications to choose from. The bulk of the market share is on the commercial side where companies such as SAP, Oracle, Salesforce, and Microsoft dominate the marketplace. There are also many open source CRMs which are usually differentiated by their programming language. Examples include:

- **PHP**: Sugar (http://www.sugarcrm.com) and vtiger (http://www.vtiger.com)
- **ASP.NET**: SplendidCRM (http://splendidcrm.com) and Tustena (http://tustena.com/crm)
- **Java**: CentricCRM (http://concursive.com) and hipergate (http://hipergate.org)

We used an open source CRM because we needed a solution that we could customize easily. We selected vtiger over Sugar as its licensing was less restrictive and it provides sales, reporting and security modules that Sugar does not.

**Communication Enabled CRM**

With a goal of making telephone and video integration as tight inside the CRM as email, we found that open source CRMs include some form of PBX integration, usually provided by asterisk (http://asterisk.org). They typically offer basic functionality such as creating single outgoing telephone sessions or popping up customer information when a call is received. Creating conference calls from within the CRM was not possible with the current implementations that we tested. Another important communication facet is text, audio and video conferencing but none of the open source CRM's we tried support these. These pain points led us to integrate direct calls, call conferencing, and video conferencing into the CRM through BigBlueButton.

In our implementation, the direct call is straight forward. All phone numbers are clickable links that create a telephone call to that customer. The conferencing call feature allows the CRM user to create a conference call with multiple customers simultaneously. The user clicks on the customers they wish to call, then the 'Conference Call' button and a conference call is created. With BigBlueButton integration,

the CRM can now start a video chat, show demos remotely using the desktop sharing feature, and provide slide shows to customers using presentation mode. These are powerful communication features that provide a higher level of interaction between the CRM user and their customers. BigBlueButton sessions can be created in the same way as conference calls and emails are created by checking the customers to contact and then clicking on the BigBlueButton button. Invitations with links are emailed to the selected customers that, when clicked, will load BigBlueButton. A popup on the CRM user's side will appear that is a clickable link to join the BigBlueButton conference.

The work that we have done is open source and the code is available for others to integrate the same communication enabled services into vtiger or port them to another CRM or another application entirely. To try out the communication services we integrated into the CRM, go to http://www.metrocave.com and use the login/password of osbr/osbr. You will need to add new customer contacts in order to test out the calling features as well as configure the user settings to use your telephone number. To configure your user settings, click on 'My Preferences' at the top right, then change the office number to use your telephone number. To add contacts that you wish to call, click on 'support|contacts' and then click on the +(create) button.

Some of the features that we wanted to add included saving the conversations in the form of audio files, converting those audio files to text with speech recognition, and then tagging that data to the customer account information inside the CRM. However, this was not possible because the Coral CEA infrastructure is not configured to save conversations and is based on a remote service that currently uses Nortel's commercial Agile Communication Environment (http://tinyurl.com/mlzgl9).

Once Coral CEA resolves this issue, adding these features will be incredibly valuable as it will allow the user to track telephone conversations as precisely as email. The more information that can be collected, the better a company can understand its customers.

**Conclusion**

We set out to integrate communications services into an open source CRM system. Our goal was to facilitate more forms of communication in an inexpensive way using Coral CEA's powerful communication services. While there are missing features to Coral CEA, such as saving audio files once a telephone call is finished, its communication service components add value to the CRM from improved customer relationships to reduced costs.

So far, our experience has taught us that at least the promise of value can be created by extending the existing features of CRM systems to include communication services. The next step is to validate our expectations of value by seeking input from our local business ecosystem. We expect there will be significant demand for a communications enabled CRM system, particularly with small companies that depend on the flexibility and reporting features that the system can provide. We hope to commercialize the work we have done by targeting a shortcoming in the open source CRM space.

*Elias Majic has a bachelor in software engineering at Carleton University. He worked for several years at software companies before pursuing a startup focused on web enabled speech recognition. He returned to graduate school to attend Carleton University's TIM program where he is currently enrolled. His thesis is focused on the adoption of speech recognition (http://www.speechapi.com).*

**University Supports for Open Access: A Canadian National Survey**

**Copyright:** Devon Greyson, Kumiko Vézina, Heather Morrison, Donald Taylor, Charlyn Black

**From the Abstract:**

The advent of policies at research-funding organizations requiring grantees to make their funded research openly accessible alters the life cycle of scholarly research. This survey-based study explores the approaches that libraries and research administration offices at the major Canadian universities are employing to support the research-production cycle in an open access era and, in particular, to support researcher adherence to funder open-access requirements. Responses from 21 universities indicated that librarians feel a strong sense of mandate to carry out open access-related activities and provide research supports, while research administrators have a lower sense of mandate and awareness and instead focus largely on assisting researchers with securing grant funding. Canadian research universities already contain infrastructure that could be leveraged to support open access, but maximizing these opportunities requires that research administration offices and university libraries work together more synergistically than they have done traditionally.

http://ojs.library.ubc.ca/index.php/cjhe/article/view/472

**Overview of Open Access Models for eBooks in the Humanities and Social Sciences**

**Copyright:** Janneke Adema

**From the Summary:**

Open Access book publishing in the Humanities and Social Sciences (HSS) is on the rise. Initiatives are emerging on an international scale, ranging from providing Open Access to single titles to full-fledged Open Access book publishers. Most of these efforts, however, are still in the experimental phase, testing and developing new publishing and business models as well as tracking customer behavior both online and offline. Nonetheless, some trends and patterns are discernable. This research has looked at a variety of initiatives and specifically at their publishing models, business models and publishing processes. Within these divisions, special attention has been paid to the nature of the content, the level of Open Access provided, the peer review and copyright policies and, finally, the strategies of collaboration.

http://www.oapen.org/images/OpenAccessModels.pdf

**Anatomy of Contemporary GSM Cellphone Hardware**

**Copyright:** Harold Welte

**From the Abstract:**

Billions of cell phones are being used every day by an almost equally large number of users. The majority of those phones are built according to the GSM protocol and interoperate with GSM networks of hundreds of carriers. Despite being an openly published international standard, the architecture of the GSM network and its associated protocols are only known to a relatively small group of R&D engineers. Even less public information exists about the hardware architecture of the actual mobile phones themselves, at least as far as it relates to that part of the phone implementing the GSM protocols and facilitating access to the public GSM networks. This paper is an attempt to serve as an introductory text into the hardware architecture of contemporary GSM mobile phone hardware anatomy. It is intended to widen the technical background on mobile phones within the IT community.

http://laforge.gnumonks.org/papers/gsm_phone-anatomy-latest.pdf

**Intellectual Property: Observations on Efforts to Quantify the Economic Effects of Counterfeit and Pirated Goods**

**Copyright:** United States Government Accountability Office

**From the Abstract:**

In October 2008, Congress passed the Prioritizing Resources and Organization for Intellectual Property Act of 2008 (PRO-IP Act) (P.L. 110- 403), to strengthen and improve the effectiveness of U.S. government efforts to protect the intellectual property (IP) of U.S. industries and IP rights holders. In the PRO-IP Act, Congress noted that U.S. IP industries have created millions of highly skilled, high-paying U.S. jobs and continue to represent a major source of creativity, innovation, economic growth, and competitiveness. The PRO-IP Act directed GAO to provide information on the quantification of the impacts of counterfeit and pirated goods on the economy and industries of the United States to help the U.S. government better protect the IP of rights holders. Our work: (1) examined existing research on the effects of counterfeiting and piracy on consumers, industries, government, and the U.S. economy; and (2) identified insights gained from efforts to quantify the effects of counterfeiting and piracy on the U.S. economy.

http://www.gao.gov/new.items/d10423.pdf

**Leveraging E-government at a Time of Financial and Economic Crisis**

**Copyright:** United Nations

**From the Description:**

The 2010 United Nations e-Government Survey: Leveraging e-government at a time of financial and economic crisis was completed in December 2009 and launched in early 2010. The report presented various roles for e-government in addressing the ongoing world financial and economic crisis. The public trust that is gained through transparency can be further enhanced through the free sharing of government data based on open standards. The ability of e-government to handle speed and complexity can also underpin regulatory reform. While technology is no substitute for good policy, it may give citizens the power to question the actions of regulators and bring systemic issues to the fore.

http://www2.unpan.org/egovkb/global_reports/10report.htm

**OSS Industry Savings**

**Copyright:** Software Improvement Group

**From the Description:**

This report investigates the use of open source software libraries in proprietary software developments. The results show that open source libraries are widely used in a set of over 300 proprietary systems, and their usage has introduced estimated savings in excess of 1 million EUR per system.

http://www.sig.eu/en/R%26D/Reports/531.html

**Guideline on public procurement of Open Source Software**

**Copyright:** IDABC

**From the Description:**

This practical guideline shows how open source software can be acquired by public agencies. It also describes how to procure software compliant to open standards. It is meant to be read by IT managers, policy makers and procurement offcers, without including too much legal detail or analysis, which are provided in an annex.

http://www.osor.eu/idabc-studies/OSS-procurement-guideline%20-final.pdf

**April 15**

Datadotgc.ca Launched: The Opportunity and Challenge

Today I'm really pleased to announce that we've launched http://datadotgc.ca, a volunteer driven site I'm collaboratively creating with a small group of friends and, I hope, a growing community that, if you are interested, may include you. As many of you already know I, and many other people, want our governments to open up and share their data, in useful, structured formats that people can actually use or analyze. Unlike our American and British peers, the Canadian Federal (and provincial...) government(s) currently have no official, coordinated effort to release government data.

http://eaves.ca/2010/04/15/datadotgc-ca-launched-the-opportunity-and-challenge/

**April 22**

Concordia University Opens its Research Findings to the World

**Montreal, QC**

Concordia University's academic community has passed a landmark Senate Resolution on Open Access that encourages all of its faculty and students to make their peer-reviewed research and creative output freely accessible via the Internet. Concordia is the first major university in Canada where faculty have given their overwhelming support to a concerted effort to make the full results of their research universally available.

http://news.concordia.ca/main_story/016711.shtml

**May 29-30**

Innovation Camp

**Vancouver, BC**

Participants will practice techniques to generate fresh ideas and implement them, create value in the real world with a team and practice challenging assumptions, negotiating, leveraging limited resources and defining success. Innovation Boot Camp is about seeing problems as opportunities and bridging the gap between action and inaction!

http://www.innovationcamp.org

---

**May 30**

Product Camp

**Toronto, ON**

ProductCamp Toronto is a collaborative, user-organized, unconference focused on product development, product marketing and product management.

http://www.productcamp.org/toronto/

**May 31-June 1**

GovCamp

**Ottawa, ON**

A number of municipalities have embraced the concepts of open government and government 2.0. There have been a number of community driven events where interested individuals have come together to progress the thinking in this area and explore tangible activities under the umbrella of change camps or citycamps. We feel that by providing an environment for a discussion at the national level, to explore the interactions between cities, provinces and the federal level, these conversations can expand and bring together all jurisdictions that support Canadian individuals and businesses. We expect that participants will explore the role of provincial and federal governments in cultivating the growth and prosperity of Canada's vibrant communities.

http://govcamp.eventbrite.com/

---

**June 7-11**

NetChange

**Toronto, ON**

Net Change 2010 brings together social causes with social tech and social networking. The week will explore tech for change: how 21st century communication is changing our society, specifically – how it is helping us address some of the world's toughest problems in new ways.

http://www.netchangeweek.ca


MASSIVE INNOVATION
Lead to Win
Exciting new businesses that are in Lead to Win now!
selectstart
CheckinWiz
media
ucreatemedia.com
catwalk networks
SCTi
EclipseSource
Castel RESEARCH
zeebumobile
Videotelephony Inc.
NOVABRAIN TECHNOLOGIES Inc.
data bridge
A NEXIA SERVICES COMPANY
EnTeraSec security
Accentway
Triple Through Sustainability
SMART ROTOR SYSTEMS
SiliconPly
EventuSix
the speed of success
cornerportal
MARKTEK
Clearview
NewEraHealth
MarkOff Security

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

| | |
|---|---|
| **June 2010:** | Growing Business |
| **July 2010:** | Go To Market |
| **August 2010:** | Interdisciplinary Lessons |
| **September 2010:** | Language Technology |
| **October 2010:** | Governance |

**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.


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.