OSBR.CA

The Open Source Business Resource

NOVEMBER
2008

NOVEMBER 2008

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

EDITOR:
Dru Lavigne
dru@osbr.ca

ISSN:
1913-6102

The length and diversity of Wikipedia's list of open source healthcare software (http://en.wikipedia.org/wiki/List_of_open_source_healthcare_software) may come as a surprise to many readers.

This issue of the OSBR provides an excellent introduction to the complexities and interoperability issues associated with healthcare software and the role open source plays in helping to resolve these issues. This month's authors also provide insight into an open source project that follows open standards, lessons learned from providing a reference implementation, the benefits of a healthcare ecosystem, and the value of open source projects working closely with standards organizations.

As always, the authors and other readers appreciate your comments and references to additonal resources. You can send these to the Editor or leave them on the OSBR website or blog (http://osbrca.blogspot.com/).

Dru Lavigne

Editor-in-Chief

dru@osbr.ca

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

The shift towards electronic health records now seems inevitable, driven by a combination of economic and demographic forces. This creates a demand for upgraded information technology (IT) infrastructure while simultaneously opening the door to new business opportunities for products that utilize that infrastructure. Whether this will result in a virtuous cycle that generates both improved public health and economic benefits remains to be seen. A key determinant will be interoperability between systems. Today, interoperability still represents a barrier to increased adoption and effective use of IT systems in healthcare – resulting in increased costs, medical error, poor care and even death.

While many of the articles in this issue of the OSBR focus on specific open source implementations, together they present an argument that interoperability in health care systems cannot simply rely on open standards but requires a widely available open source platform. The emergence of constructive partnerships between standards experts and open source implementers has the potential to accelerate the adoption of electronic health records and create an ecosystem of cooperating private and public sector players.

Brian Barry, CEO of Bedarra Research Labs and CTO of Open Health Tools (OHT), covers OHT's planned development of a common platform. In his article, he applies his experience with the founding of Eclipse to the problems of interoperable healthcare.

Mark Yendt, Duane Bender and Brian Minaji of Mohawk College describe their reference implementation of the Canada Health Infoway pan-Canadian Electronic Health Record Solution. They see the need to involve a range of partners from both the private and public sector to achieve success for the project.

Jacob Brauer of WebReach describes the Mirth Project, an open source healthcare interface engine and interface repository created and professionally supported by WebReach. Mirth attacks the interoperability issue directly through providing an interconnecting middleware for health information systems.

Skip McGaughey, Executive Director of OHT, and Ken Rubin, a senior healthcare architect, describe the requirement for a more out-reaching community which encompasses end users, operational users, and developers.

Jon Siegel and Richard Mark Soley explore the relationship between open source and open standards by examining the results of their interviews of developers of open source tools based on open standards.

Throughout these papers we see mention of the essential role played by healthcare data standards. These are clearly necessary for the development of any interoperable electronic health record solution.

Brian Barry is CEO of Bedarra Research Labs and CTO of Open Health Tools. From 1991-2002 he served variously as Chief Scientist, CEO, President and CTO at Object Technology International, Inc. Under his leadership, OTI developed the Eclipse Platform and the IBM VisualAge family of products. Dr. Barry has published a number of research papers and articles on a wide variety of technical subjects. He has served on the Program Committees for software conferences such as OOPSLA, ECOOP, AOSD and Agile Development, was a coauthor of the ANSI Smalltalk standard, and actively participates on research review boards and committees.

Peter Tanner divides his time between being a Retiree-on-Call at IBM Canada, and handling the intellectual property policies and strategies for Open Health Tools. As Director of Business Development at Object Technology International, Mr. Tanner was directly involved with the legal and business issues during the founding of Eclipse.

"People developing software don't understand the complexities of health care, and the health care professionals don't understand what it takes to produce that software. It's hard for them to understand each other, so [one major benefit of OHT] will be to bring them together in one room."

http://www.govhealthit.com/online/news/350300-1.html

Healthcare has been characterized as a multi-trillion dollar cottage industry. It is highly fragmented, labour intensive, barely connected, extremely competitive, and has many different vendors and proprietary solutions. The rising cost of healthcare is straining budgets at all levels of government and imposing financial burdens on corporations and individuals alike. Against this backdrop, legitimate concerns about privacy have led to a plethora of regulations requiring complex administrative, physical and technical infrastructure to safeguard sensitive health information. Governments are attempting to impose standards and specifications from the top down to improve efficiency in healthcare delivery. These standards are broad, complex and, for the most part, lack implementations. In short, things are in a bit of a mess.

A consensus is emerging around two initiatives that promise to improve the current situation. The first is to foster widespread adoption of Electronic Health Records (EHR, http://en.wikipedia.org/wiki/Electronic_health_record). The second is to improve accessibility and interoperability between EHR systems. In this article, we present Open Health Tools (OHT, http://www.openhealthtools.org), an open source ecosystem where members of the health and information technology (IT) professions can collaborate to build interoperable EHR systems.

Open Health Tools

Interoperability problems and their resulting inefficiencies can result in increased costs, medical error, poor care, and even death. There are studies showing that lack of immediate access to patient care records results in thousands of deaths every year (http://jama.ama-assn.org/cgi/content/full/293/5/565). In the U.S. alone, it is estimated that 200,000 people die every year from medical errors, many of which would be preventable if we had connected, interoperating EHR systems (http://www.healthgrades.com/AboutUs/index.cfm?fuseaction=mod&modtype=content&modact=Media_PressRelease_Detail&&press_id=135).

While interoperability is important, progress towards interoperable EHR systems remains painfully slow. Former Intel Chairman Andrew Grove compares healthcare with the mainstream IT industry: "When it comes to operational efficiency, nothing illustrates the chasm between the two industries better than a comparison of the rate of implementation of electronic medical records with the rate of growth of e-commerce" (http://jama.ama-assn.org/cgi/content/abstract/294/4/490). While IT offers huge opportunities to improve healthcare systems and the quality of care, it also presents us with enormous challenges.

OHT is an international open source organization that was formed in 2007 to accelerate the implementation of EHR systems and promote interoperability. OHT was incubated within the Eclipse Foundation (http://www.eclipse.org) and Health Level 7 (http://www.hl7.org). At the time of writing, it has over 30 members representing a cross-section of the healthcare community from Australia, Canada, the U.S., the United Kingdom, and continental Europe.

The members of OHT believe that open source can be a powerful lever to expedite EHR deployment, improve interoperability between systems, and accelerate the delivery of the attendant social and economic benefits. They subscribe to the basic principles underlying open source development: i) contributors primarily work to satisfy their own requirements; ii) software is contributed to a common pool when it makes business sense to do so; iii) development effort is coordinated by senior developers and architects who ensure that the end result is timely and coherent; and iv) all participants share in and leverage the final result.

Many open source efforts target healthcare. However, none has gained significant traction. We believe there are reasons why OHT will succeed where other efforts have not:

1. OHT has brought together national health agencies, vendors, care providers, standards organizations, academics, researchers, and payers.

2. OHT has a very clear sense of its primary mission to develop and deliver running software.

3. OHT has no internal confusion about acting as a quasi-standards organization (although many of the key standards groups are members) nor is it a vendor association promoting members' products (although many vendors are also OHT members).

4. OHT understands that its ecosystem needs to balance the social benefits of making code freely available with the business necessity of helping its commercial members generate profits – a portion of which then can directly or indirectly flow back into OHT.

This last point is the key to long term sustainability. We encourage the use of commercial friendly processes and licenses such as the Eclipse Public License (EPL, http://www.opensource.org/licenses/eclipse-1.0.php) and help members promote products and discover synergies and new opportunities. We believe that this sets OHT apart from most open source healthcare efforts which don't see it as their business to promote commerce and tend to use viral licenses such as the Free Software Foundation's GPL or LGPL under the misguided notion that this will prevent "misuse" of their software by commercial entities. What these licenses do in practice is destroy opportunities to create value-added products by making it difficult to combine open source and proprietary software in the same package.

The Quest for Interoperability

Proprietary boundaries are largely responsible for the current situation of non-interoperable EHR systems. Different proprietary systems have different information architectures. Current standards such as HL7 do not completely disambiguate EHR structures, even when implementations claim to be compliant. It is quite possible, in fact it is typical, to have HL7 compliant but non-interoperable implementations. With care providers, payers, patients and healthcare professionals all demanding action, the path of least resistance for most vendors is to propose incremental solutions that paper over the cracks between current implementations with "least common denominator" integration approaches. Gateways are a typical response: those parts of the record structures that can be mapped between two EHR implementations will be, but some information is normally lost because the two information architectures cannot be completely aligned. It is easy to see that things just get worse when health records are moving across several proprietary boundaries.

The result is usually primitive interoperability at the expense of functionality. This approach only perpetuates the problems. A real solution requires: i) a common platform that can be shared by all and which is built on strong and proven integration models with well defined interfaces; ii) powerful data transformation capabilities; and iii) a base of unifying underlying technologies. To be successful, interoperability needs to be designed in and developed from the platform up to the application, not imposed top down from the proprietary application to a gateway or adapter.

What about standards organizations? Won't more and better standards lead to interoperability? Unfortunately, as experience over the last twenty years has shown, the inherent complexity of the healthcare domain and the dynamic nature of its logical record structures (with new information types being continually added or modified as medical practice and technology evolves and changes), make it very unlikely that this issue will ever be completely solved by standards efforts.

To be clear, no one is saying that standards are not worthwhile. The point is that EHR interoperability is far too complex to be solved by standards alone. Moreover, there are many cases where deeper levels of integration other than simply exchanging records are desired. In many instances, much more efficient and capable processes could be realized at lower cost if it was possible to integrate systems at the application logic level. This is the promise offered by service oriented architectures (SOA, http://en.wikipedia.org/wiki/Service-oriented_architecture) which expose facilities as composable services grouped logically by function.

The Need for a Common Platform

The possibility of achieving a common ubiquitous platform will engender some natural skepticism. OHT's motivation for this goal comes from the shared history of its founders in the Eclipse Project. The Eclipse platform was designed to address the interoperability needs of the software development tools market. Before Eclipse, this market was populated by hundreds of different tools which interoperated (although not well) by sharing files and using a few actual and de facto standards. Microsoft had the most capable platform but its growth was stalled. No single competing vendor--not even big players like IBM, Borland, and Rational--could afford the investment required to build a broadly capable base product and then add all of the extensions, customizations, and specializations required to satisfy the breadth of customer requirements. The result was customer dissatisfaction and inefficiency, combined with a certain fatalism that this was the way the industry had always been, and nothing could be done about it. This is very reminiscent of attitudes seen in the healthcare industry today.

Eclipse offered a well-engineered shared infrastructure, shifting the focus for investment and innovation away from building (and re-building) the same base functionality. Instead, vendors could collaborate to build the platform and then compete on added value products. By sharing common infrastructure, they could concentrate their resources on climbing the value chain to build hundreds of differentiated products, all sharing the same base components. As an added bonus, since they shared the same platform, it was relatively inexpensive to build interoperating products. This economic change altered the business models and made interoperation a winning strategy in most cases.

We believe Eclipse succeeded because the infrastructure was designed from the ground up to support interoperability. Due to its rich set of high quality components and the economies of scale, migrating to Eclipse was the most cost-effective option long term for most vendors. In effect, the commercial value proposition was changed so that vendors could share a common infrastructure, but compete on value-added products built with “Eclipse inside”. We are firmly convinced that Eclipse succeeded because it was freely available and not controlled by any single dominant industry player. For an interoperability platform to achieve its purpose, it must be (nearly) ubiquitous. Applications interoperate because they are all building upon the same base. An open source solution works well for the base; in fact, it may be the only viable solution. The platform, much like the national highway system, must become a shared resource, owned by no one, and enabling everyone to utilize the same level playing field. This creates economic value and addresses customer needs. OHT members believe it may be possible to achieve the same result in the healthcare market, and that OHT is the right vehicle to advance this agenda. Based on the experience that the OHT team has with Eclipse and with EHR implementations and standards, we see three key “grand challenges” as the prerequisites for success:

1. OHT needs to develop a high quality modular OHT platform that is component-based and has built-in integration, extension, and customization mechanisms. In other words, interoperability needs to be designed in.

2. OHT must build a development and user community that includes a representative cross-section of healthcare stakeholders. Participation by national health agencies, standards organizations and key vendors is critical.

3. OHT must create a self-sustaining ecosystem that will allow the OHT platform to continue to grow and evolve as new needs become apparent. This is very much a culture-changing mission within the healthcare community.

Pursuing these goals presents a challenge. Moving beyond the current situation to a common shared platform will be a difficult, multi-year endeavour requiring more than just a superior technical solution. It will also involve a significant community building effort and the creation of a supportive socio-economic ecosystem which is designed to leverage commercial forces and not fight them. A tough challenge certainly, but previous experience with the Eclipse platform and the Eclipse Foundation shows that it is not impossible. And the stakes are high, since interoperability ultimately has the potential to lower costs, save lives, and improve care.

OHT Platform

The OHT platform is an enterprise service bus for services that implement healthcare applications; this is often called a “health services bus” or “health services spine”. While the exact shape of the OHT platform is still a work in progress, OHT members are working to provide a more precise definition and implement parts of the system. The planned services to be implemented are:

Infrastructure services: include security and privacy, patient and provider registries, communications, medical device integration, and workflow and business rules.

Patient information services: include record location and management, entity identification, distributed data access (CRUD, http://en.wikipedia.org/wiki/Create,_read,_update_and_delete), indexing, and replication.

Interoperability services: include data interchange, legacy adapters, and data transformation.

Terminology services: include administration, search and query, authoring and maintenance, and concept/terminology mapping.

Analysis services: include reporting, analytics, and data warehouse.

Public health services: include outbreak management, detection and notification, geospatial mapping, and visualization.

The OHT platform will provide built-in extensibility mechanisms to enable users to create first class extensions. Potential applications that the OHT Platform is targeted to support include EHRs, personal health records (PHRs), pharmacare, laboratory, radiology/imaging, and viewers/portals for patients.

As a practical matter, no platform will be successful without a complementary set of supporting tools. OHT tools projects planned or under way include:

Modeling: healthcare artifacts, clinical content, and medical data.

HL7 messaging: message modeling and design, message instance editors, and message example generators.

Terminology: design, update, maintenance and deployment.

Conformance and test: profile management, test design, test generators, simulators, and test execution.

IHE profiles: implementations of profiles defined by Integrating the Healthcare Enterprise (IHE, http://en.wikipedia.org/wiki/IHE), an industry backed initiative to improve interoperability.

The OHT platform provides standard domain-aware interfaces and reusable software components that can be assembled into patient-centered services and applications. It provides component and programming models that enforce constraints on how application software is developed. The result is applications that interoperate seamlessly because they are running the same code. Interoperable applications are cheaper to build due to the use of free, high quality pre-existing code. Users find such applications easier to learn and use because they are based on similar concepts and interaction models. Operator skills acquired on one application are transferable to another.

Creating the OHT Community

As noted above, it is critical to create both a platform and a supporting community or ecosystem. These need to evolve in tandem as the platform and the ecosystem share a symbiotic relationship. In order to establish a community, there needs to be a common shared platform on which to build. Without a continuing research and commercial ecosystem, the platform ceases to innovate, becoming obsolete and losing relevance over time. Establishing the platform and the ecosystem requires a substantial initial investment which typically comes from public sources or from an entity which stands to gain in the long run from the existence of the platform and the community.

After an initial bootstrap period of three to five years, the ecosystem must generate sufficient ongoing economic and social benefits to become self-sustaining. The best way to ensure this is to encourage commercial activity around the platform early, so that vendors can realize a return on investment in the platform and the marketplace identifies which enhancements and value added products offer real value.

There must also be a mechanism for continuing public investment to recognize and encourage innovations and applications that have social value--such as public health--even if there is no commercial return on investment (ROI).

Conclusion

The principle contributions of OHT will be:

1. Open source software projects to produce the OHT platform and supporting software tools.

2. A supporting commercial ecosystem built around the OHT open source code base that will create long term sustainability.

3. An open forum and level playing field where providers, vendors, standards experts, caregivers, and software developers can collaborate and share assets and expertise.

The end result of deploying health IT systems based on OHT software will be improved care, better safety and lower costs. The immediate beneficiaries will be patients, public healthcare providers such as state operated national health agencies, and private payers such as insurance companies. Secondary beneficiaries include vendors who will create and exploit new markets and cut development costs, private providers who can adapt their business models to exploit lower cost open source solutions, and physicians who can deliver more patient-centered care.

Brian Barry is CEO of Bedarra Research Labs and CTO of Open Health Tools. From 1991-2002 he served variously as Chief Scientist, CEO, President and CTO at Object Technology International, Inc. Under his leadership, OTI developed the Eclipse Platform and the IBM VisualAge family of products. Dr. Barry has published a number of research papers and articles on a wide variety of technical subjects. He has served on the Program Committees for software conferences such as OOPSLA, ECOOP, AOSD and Agile Development, was a co-author of the ANSI Smalltalk standard, and actively participates on research review boards and committees.

Recommended Resources

Open Health Tools: Architectural Vision
http://www.openhealthtools.org/Reports/Apr08/OHTArchitectureVisionV3-1.pdf

Commission on Systemic Interoperability: Ending the Document Game
http://endingthedocumentgame.gov

Canada Health Infoway
http://www.infoway-inforoute.ca/en/home/home.aspx

Health Care Renewal in Canada : Accelerating Change
http://www.healthcouncilcanada.ca/docs/rpts/2005/Accelerating_Change_HCC_2005.pdf

"We're not in the healthcare business; we're in the information management business. We should start thinking as information managers dealing with healthcare information, and think about the tools we need to do it properly."

2015: Advancing Canada's Next Generation of Healthcare

Developing a reference implementation of the Canada Health Infoway pan-Canadian Electronic Health Record Solution (EHRS, http://www.infoway-inforoute.ca/en/WhoWeAre/Overview.aspx) standard can be a useful step in ensuring the successful and cost-effective development of full scale electronic health systems in the Provincial Ministries of Health across Canada. These jurisdictions could benefit from the knowledge gained and the artifacts created in this prototype environment. The reference implementation utilizes an Enterprise Service Bus (ESB, http://en.wikipedia.org/wiki/Enterprise_service_bus) architecture and a Service Oriented Architecture (SOA, http://en.wikipedia.org/wiki/Service-oriented_architecture) design approach to build a Health Information Access Layer (HIAL), as recommended by Canada Health Infoway. The system components and supporting technology developed will be released as open source. This set of technology could represent a starting point for prototyping an implementation in a production environment, for creating a standards development platform, for standards conformance testing, and/or as a test bed for evaluating alternative software components in a HIAL environment.

Mohawk Applied Research Centre for Health Informatics (MARC HI, http://www.mohawkcollege.ca/marc/hi/) at Mohawk College, along with public and private sector partners, is continuing to build a reference implementation of the pan-Canadian Infoway standard that demonstrates the ESB/SOA approach.

This article summarizes the project to date and suggests future research areas that will reduce the cost, risk and time barriers to widespread adoption of eHealth systems in Canada.

An Overview

Developing an open source reference implementation of the Canadian EHRS can be a useful step in ensuring the successful and cost-effective development of full scale systems in the jurisdictions across Canada. A project to build a reference implementation of the pan-Canadian HIAL was undertaken at MARC HI to demonstrate the feasibility of an SOA design approach and to assist in the development of a pool of knowledge around the applicable technologies, including orchestration service products, database products and a virtual machine environment.

The reference implementation was built on the pan-Canadian EHR Infostructure standard which in turn is based on Health Level 7 version 3 (HL7v3, http://en.wikipedia.org/wiki/HL7#HL7_Version_3), managed by Canada Health Infoway. The standard describes a reference information model (RIM, http://www.hl7.org/Library/data-model/RIM/modelpage_mem.htm) consisting of low-level data types, medical concepts and full healthcare interactions. The Canadian implementation is a restricted subset of this international standard. A number of countries have started to use this standard as a basis for electronic healthcare records. Within Canada, all conforming systems must support the standard and the implemented interactions to ensure jurisdiction to jurisdiction interoperability.

The goals of this reference implementation are to:

- build a set of reference interactions that illustrate the process of making the EHRS operational

- evaluate commercial and open source tools that can assist in the development process
- develop tools and methodologies that will assist in the creation of full-scale production systems
- disseminate information by making available the source code for the tools developed and making available the design model of the reference implementation
- assemble a group of software engineering and healthcare professionals for the purpose of gaining knowledge and experience in these new tools and techniques

Point of Service (POS, http://en.wikipedia.org/wiki/Point_of_service) applications will need to develop interfaces to the jurisdictional HIALs to support interactions of concern. For example, lab systems will need to be developed or modified to support the standard lab order and result interactions. During the development process it was determined that POS applications will have difficulty in building the appropriate interfaces to the HIAL given the complexity of the messages. A current priority of this project is to build a focused domain based API that will hide the complexities of the messages to reduce the barriers of POS integration with the HIAL.

Project Stakeholders

A number of participants have become active stakeholders in the project:

Mohawk College: Mohawk provides a home for the project, including: i) key development/design leadership; ii) information technology (IT) resources for hardware and software; and iii) administration, management and governance services. Mohawk will benefit from its participation in the project by: i) enabling faculty and students to participate in a leading-edge IT project; ii) developing partnerships with leading private and public organizations in the eHealth marketplace; and iii) creating an applied research foundation for a possible diploma program in health informatics.

Canada Health Infoway: the project scope and direction is guided by the pan-Canadian EHRS blueprint developed by Infoway. Infoway will also provide guidance through the activities of its Standards Collaborative (http://www.infoway-inforoute.ca/en/WhatWeDo/SCOverview.aspx) and through its ongoing involvement on the project steering committee. Infoway will benefit from its participation in the project by: i) tracking successes/failures of the project and feeding lessons learned back into the pan-Canadian eHealth standards development process; ii) leveraging fact-based metrics regarding the performance of the reference implementation to encourage adoption of the pan-Canadian standards by its partner jurisdictions; iii) affording vendors an opportunity to test new product developments and integration adapters against a standards-compliant reference implementation; and iv) affording partner jurisdictions an opportunity to prototype standards-compliant systems using the reference implementation as a basis for their regional EHRS initiatives.

Private sector partners: the project cannot succeed without the participation of private sector partners. They will provide direction through their participation on the project steering committee and necessary financial and in-kind support (hardware, software, training) to the College. Private sector partners will benefit from their participation in the project by: i) developing demonstrable, hands-on expertise regarding the pan-Canadian EHRS blueprint and its companion standards; ii) developing or configuring proprietary companion product and service offerings that are demonstrably compliant with Infoway's EHRS blueprint and standards; iii) being able to communicate evidence-based assertions regarding compliance and/or performance characteristics of their product offerings in the context of the pan-Canadian EHR; iv) cultivating relationships with other leading private and public sector organizations in the Canadian eHealth market space; and v) building brand as a leading-edge organization involved in state-of-the-art EHRS development.

EHRS jurisdictions: in Canada, an EHRS is provided or hosted by a jurisdiction and the end-users are the healthcare providers in the province, territory or region served by the local EHRS. These two tiers of customer are represented on the project advisory board. These stakeholders benefit from their participation in the project by: i) providing guidance regarding which project activities will most directly address the pressing needs of jurisdictions and care providers; and ii) being able to reduce their implementation risk by basing EHRS pilot project and/or procurement decisions on technologies and IT approaches that have been successfully prototyped in the college's development environment.

Prototype Development

This project uses an iterative approach of design/build/refine/repeat to implement the pan-Canadian blueprint for the EHRS. An Agile development process (http://en.wikipedia.org/wiki/Agile_software_development) was used with the focus of getting a limited functional system up and running as soon as possible. The Agile process is ideal when the team is small, the project has a very focused set of deliverables, and the outcome will be used as a starting point for further development. This approach was selected as an ideal fit given the initial set of conditions of the project as stated below:

Tight time frame. Initially the project was given a four month period to demonstrate feasibility. Creating a functional sub-set of interactions was the prime focus.

Limited resources. The project started with a single university faculty member with two part-time senior third year students.

Limited knowledge of domain. The original staff allocation on the project was one software engineering faculty and student developers who had limited knowledge of the healthcare domain. The focus of the initial development was limited to patient discharge interactions.

The interactions in an interoperable EHRS consist of PUT, GET and LIST interactions. PUT interactions require that data about a discharged patient be placed into the data repository. LIST operations retrieve summary information based on specific queries to the EHRS. GET transactions are used to return detailed information around a specific transaction.

A limited physical architecture was deployed to support the prototype development. It consisted of three Intel based server class computers running enterprise based virtual machine management software. The selection of a virtual machine hosting environment was selected to allow flexibility in the construction of the required underlying implementation, which has grown to approximately 40 servers.

The software design consists of an ESB implemented through the use of orchestration engine software. Each of the components in the HIAL has been set-up on a separate virtual machine instance, usually with a web-service which exposes the functionality.

Each of these services communicates to a dedicated data repository or external service within or outside of the HIAL. The design allows for development of alternative implementations for a particular component of the HIAL and a straightforward mechanism for swapping components. Microsoft BizTalk software was selected as the orchestration software since access to this software was available through an MSDN college license. The initial web services were developed in Visual Studio .Net 2005. Phase 1 of the project used this infrastructure to complete the discharge interactions in early January 2008.

Creating a Demonstration System

Upon successful demonstration of the patient discharge interactions, the project was expanded to include a private sector development partner (Satyam Computer Services Limited) who contributed an offshore development team and financial resources to the project. The team at MARC HI was also expanded to include a second computer systems faculty member, two recently graduated developers from the Computer Systems Software Engineering Technology program, and three part-time students from the same program. This newly expanded team was used to do the following:

- complete a set of transactions for patient referral
- create a simple POS application for discharge and referral
- create a visualization tool to illustrate the HIAL interaction processes
- develop a set of transactions around health system management

The health system management project was undertaken with the Canadian Institute for Health Information (CIHI, http://www.cihi.ca/).

CIHI is an independent not-for-profit organization that provides analysis of healthcare information for Canadians. Currently, this analysis requires the collection of de-identified data from many disparate agencies and systems. Performing this task requires significant effort. The project with MARC HI provided a demonstration of a process that could both speed up the access to data and remove the requirement for manual transcription.

In May 2008, the working prototype from MARC HI was demonstrated at the 2008 Canadian eHealth Conference in Vancouver, BC. Infoway, CIHI, Oracle Corporation, ORION Health and a number of industry participants contributed to the demonstration. This was the first time a fully functional HIAL had been constructed and demonstrated based entirely on the Infoway blueprint. The demonstration of the system has been regarded as a success by those who participated in the EHRS interoperability showcase.

In July of 2008, MARC HI and Infoway made a joint charter project submission to the Open Health Tools (OHT, http://www.openhealthtools.org/) consortium. Upon final acceptance of the submission, the source code and tooling developed will be released through OHT as open source software.

Project Discoveries

The national strategy of Canada is to move to electronic medical records by implementing the Canada Infoway model. This project has made a number of discoveries during the initial application development. The most significant discoveries are listed below:

Design process: the Agile process was ideal for the initial development of discharge interactions.

A small team that communicates frequently (almost continuously) and has a focused scope can complete a challenging project. Using the process with the off-shore team required refinement as time-zone issues and reduced face-to-face interaction proved to be challenging. The development process evolved into a more traditional model where the design specifications were developed on-site at MARC HI and implemented by the off-shore team. Again, frequent communication was required to keep the development process on track.

Tooling: the current tools available for manipulation of HL7v3 messages are limited and difficult to use. The messages are complex from both a modeling and implementation viewpoint. From a modeling perspective, the messages are designed to minimize ambiguity of information. This goal has been recognized as a key component to providing a framework that can ultimately provide full interoperability. However, this design approach has resulted in a complex message structure. The HL7v3 implementation technology specification (ITS) as selected by HL7 is XML (http://en.wikipedia.org/wiki/Xml). The XML has been designed with a full set of schemas that describe the content requirements of a valid XML message. These schemas are very complex since they have been designed to describe completely all possibilities for the transaction. From a developer perspective, this can be intimidating upon initial inspection. Current automation tools have limited success with the schemas and some tools fail completely. In terms of timeliness and in producing code that is useful to the developers, the current state is less than optimal.

Limited application programmer interfaces (APIs): one of the difficulties with the HL7v3 XML/ITS is the level of complexity. Existing automation yields an extremely cumbersome object model which consists of many levels of encapsulation. A domain focused API could solve this issue, but to date the APIs developed have been too abstract to provide this advantage.

Experienced developers: the lack of sufficient trained software developers is also proving to be a significant issue in rolling out full-scale production systems. The reasons for this include: i) XML and web-based services architecture technologies are relatively new, appearing within the last 5 years; ii) demand for well trained developers exceeds the current supply and will for the foreseeable future; and iii) enterprise based software for developer training is not widely available and can be expensive for small sized developer organizations and training institutions. In short, few organizations are investing sufficiently in the infrastructure (people and systems) to support the required development effort.

Future

Two areas have been identified to offer the most promise to the future of the interoperable EHRS:

1. Domain API. The design of a domain focused API could significantly ease the building of new applications and at the same time reduce the barrier to existing developers in the healthcare domain. The main goal of this type of API would be to shield the developers of POS applications from the complexities of the XML ITS. Creating an open source solution that could form the basis of extended and customized solutions in the commercial space is currently of prime interest to this project.

2. Training. The area of ESB and web services within the healthcare domain requires the education of developers and architects. Specific courses to bring current developers up-to-speed in the technology is an area that requires significant investment and effort. In addition, students being currently educated in computer systems programs need the opportunity to include optional courses in the healthcare domain to enable them to provide value to future employers. The need for new programming talent in this area has been identified as a significant risk to successful full-scale implementation of interoperable EHRS in Canada.

Conclusions

The EHRS project at MARC HI has accomplished a great deal in its initial phase. The demonstration of a successful system has instilled confidence in the development community that the Infoway design can be implemented as specified. A number of challenges remain that will be the focus of future research. The prime focus is the development of an appropriate API that will reduce the barriers to full-scale production implementation.

Recommended Resources

Information Governance of the Interoperable EHR
http://www.infoway-inforoute.ca/Admin/Upload/Dev/Document/Information%20Governance%20Paper%20Final_20070328_EN.pdf

2015: Advancing Canada's Next Generation of Healthcare
http://www.infoway-inforoute.ca/en/pdf/Vision_2015_Advancing_Canadas_next_generation_of_healthcare.pdf

Mark Yendt is a fulltime Faculty member in the Software Engineering Technology Department at Mohawk College. He graduated from the Chemical Engineering Technology Program at Mohawk College in 1983. Mark worked in the wastewater engineering industry with a focus on the development of process modeling software. In 1994, Mark was a co-recipient of the Harrison Prescott Eddy Medal for Noteworthy Research awarded by the Water Environment Federation. His current research activities include software process visualization and optimization of systems using the HL7v3 standard.

Duane Bender is a licensed Professional Engineer (P.Eng.) in Ontario and a fulltime Faculty member in the Software Engineering Technology Department at Mohawk College. He graduated from the Computer Engineering and Management program at McMaster University in 1996 and is currently enrolled in the McMaster MBA program. Duane is the founding faculty member of the MARC HI, and performs Applied Research in the area of eHealth. His current research interests include building large scale eHealth systems using ESB and SOA architectures and the IHE, CDA, DICOM and HL7v3 standards.

Brian Minaji graduated from Computer System Technology at Mohawk College in 1986. He worked as an IT professional in a number of different industries for the next 15 years. In 2001, he joined the faculty of Software Engineering Technology at Mohawk College. He has been involved with the MARC HI project since its inception. His research interests include helping build the Canadian EHRS Reference Implementation.

"Open source operating systems, databases, web servers, and system utilities are slowly making inroads into hospital data centers that formerly housed only proprietary technology products. Add another category in which an open source alternative is available: integration engines."

Tim Dotson
Inside Healthcare Computing

The Mirth Project (http://www.mirthproject.org) is an open source healthcare interface engine and interface repository created and professionally supported by WebReach (http://www.webreachinc.com). Mirth provides standards-based tools to develop, test, and deploy interoperability solutions for healthcare information systems and information exchanges.

This article provides an overview of healthcare interface engines. It discusses Mirth and the healthcare and connectivity standards it supports. Lastly, the article compares Mirth to other interface engines.

Introduction to Healthcare Interface Engines

Healthcare interface engines, also known as healthcare integration engines, solve the problem of sharing and exchanging data between healthcare applications. Data interchange is a significant problem in healthcare. There are numerous vendors, data providers, and custom applications that need to exchange information using evolving standards. To make things worse, many legacy healthcare applications do not support a standard, yet they are required to intercommunicate with other standards-based applications. Healthcare interface engines connect applications by mapping and transferring data between the applications using standards and data definitions understood natively by each application.

Interface engines have been available for many years and there are many engines available in the market. The cost of proprietary engines range from the low hundreds of dollars to hundreds of thousands of dollars for organization-level licenses. A review of interface engines is available from KLAS (http://www.klasresearch.com), an independent market intelligence research firm. The KLAS Interface Engines Market Review collects data about leading interface engines and ranks them by several criteria. This article will discuss an open source interface engine called Mirth. Even though Mirth is not ranked by the KLAS review, many people in the Mirth community claim Mirth is functionally equivalent to high-end proprietary interface engines.

Introduction to Mirth

Mirth is middleware that connects health information systems so they can exchange clinical and administrative data. Mirth is released under the Mozilla Public License v1.1 (http://www.opensource.org/licenses/mozilla1.1.php) and is professionally supported by WebReach, a Health information technology (IT) solutions company based in California.

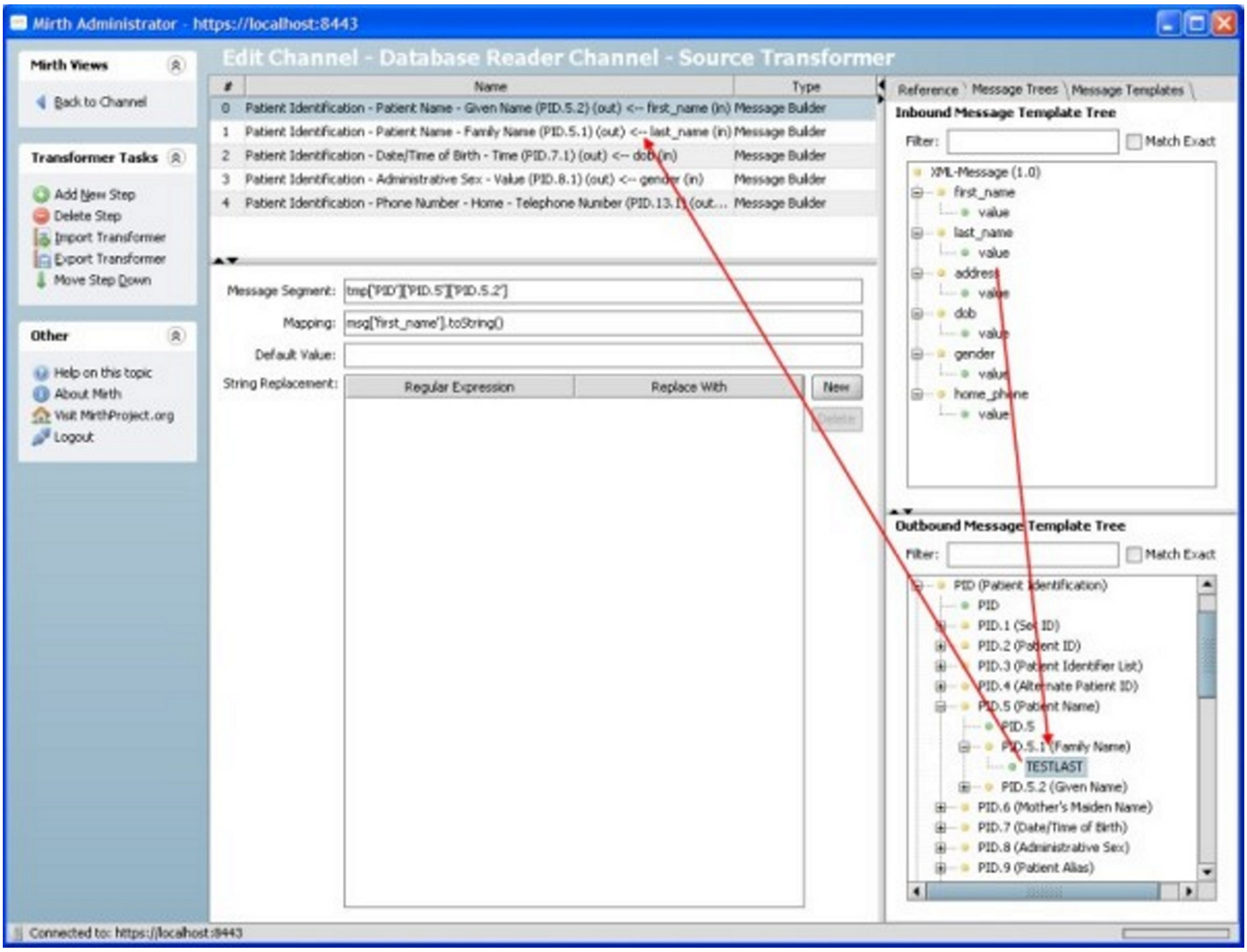
There are many standards in healthcare, with a diverse range of protocols and types of data. There are different health information systems such as labs, pharmacies, clinics, hospitals, and many others. Each of these systems might have different protocols, mismatched versions, and incompatible data. Some systems might actively use HL7, X12, and DICOM images, while others simply have a database to read from or communicate with XML and comma separated values. Add to that the lack of control administrators have over current and legacy applications, and the healthcare interoperability problem becomes apparent.

This is where Mirth steps in as the easy to use and deploy middleware solution. Mirth can lie between any number of health information systems, whether they speak a standard healthcare language or not, and help them communicate.

Mirth is a flexible health IT infrastructure component and can serve many roles. It can provide central integration exchange at a hospital, an information gateway for a clinic or reference lab, or an information exchange for a Health Information Exchange (HIE, http://en.wikipedia.org/wiki/Health_information_exchange) or Local Health Integration Network (LHIN, http://en.wikipedia.org/wiki/Lhin). It can also act as the integrated interface engine for an electronic health record (EHR, http://en.wikipedia.org/wiki/Electronic_health_record) or as an extract, transform, and load (ETL, http://en.wikipedia.org/wiki/Etl) tool.

By using a point-and-click interface and JavaScript to map data elements, Mirth speeds the development of interfaces for secure exchange of data across formats. For example, one can exchange data from a delimited file to HL7 or vice versa. This ease-of-use reduces barriers to the formation of health information exchanges involving diverse information systems and advances initiatives aimed at improving patient safety and continuity of care. Figure 1 provides a screenshot of this interface. In this example, Mirth is receiving input from a database and has converted the data to HL7 format:

Figure 1: Database Reader Channel - Source Transformer



The Mirth Reference Guide (http://www.webreachinc.com/wiki) provides many more examples and screenshots for using Mirth.

Mirth also provides an open framework and repository for creating and sharing Mirth channels (http://www.mirthproject.org/index.php?option=com_docman&Itemid=43). A Mirth channel is an interface that routes, filters, and transforms messages from one source to one or many destinations. These channels can also be chained together for more complex logic. Sharing channels enables those in the healthcare IT community to benefit from the work of others and eliminates the redundancy inherent in current processes that require each organization to develop the data mappings between systems. An ideology that focuses on sharing and openness is a major leap forward over the closed and proprietary system the healthcare IT community is used to. This community-based, open approach has been demonstrated to speed innovation in other industries and is now available for the first time in healthcare.

The interoperability problem is universal and, as a result, Mirth has a global community of end-users and contributors. Mirth has been successfully used in production at hundreds of facilities. Below is a sample of some of Mirth's professionally supported customers:

- Interior Health, Vancouver
- Infection Prevention and Control Program, Calgary Health Region
- HealthBridge, Ohio
- Indiana Health Information Exchange, Indiana
- Redwood MedNet, California
- MedSphere, California
- Epocal, Ottawa
- San Joaquin General Hospital, California
- Tarrant County Health APC, Texas
- Silver Cross Hospital, Illinois
- VA, Pennsylvania
- NHS, UK

Supported Healthcare and Connectivity Standards

Mirth supports healthcare standards such as HL7 (http://en.wikipedia.org/wiki/HL7) for the exchange, integration, sharing and retrieval of electronic health information, DICOM (http://en.wikipedia.org/wiki/Dicom) for medical imaging, X12 (http://en.wikipedia.org/wiki/X12) for the transmission of electronic data, NCPDP (http://en.wikipedia.org/wiki/NCPDP) for pharmacy data, and XML (http://en.wikipedia.org/wiki/XML) to facilitate information flow for lab results, medical records, radiology data, transcription information, claims data, and so on. These standards are further described below.

Supported healthcare data standards include:

HL7 v2 & v3: Health Level 7 is a widely recognized standard for exchanging healthcare information between healthcare applications and systems. More information can be found at http://www.hl7.org/.

DICOM: Digital Imaging and Communications in Medicine is a standard that supports messaging and imaging between imaging devices and systems. More information can be found at http://medical.nema.org/.

NCPDP: Mirth supports the National Council for Prescription Drug Programs Script standard which facilitates ePrescribing (http://en.wikipedia.org/wiki/Eprescribing). More information can be found at http://www.ncpdp.org/.

EDI X12: Mirth supports the common transaction sets for healthcare and insurance. More information can be found at http://www.x12.org/.

XML: XML is Mirth's native format and Mirth has robust XML support including support for XSLT.

Mirth also provides robust support for the following connectivity standards and protocols:

* JDBC

* SMTP

* Samba

* Delimited file such as CSV

* FTP and SFTP

* HTTP/HTTPS

* JavaScript Custom Connector

* JMS

* MLLP

* PDF and RTF

* SOAP and TCP

Project Comparison

Although Mirth is in the very specific domain of open source software (OSS) integration engines that focus on healthcare, there are other OSS integration engines in the same domain.

JEngine (http://jengine.org) is another open source enterprise integration engine implemented in Java. JEngine is deployed inside the JBoss (http://en.wikipedia.org/wiki/Jboss) application server, and uses XML configuration files and BeanShell (http://en.wikipedia.org/wiki/Beanshell) in order to set up interfaces. The latest version of JEngine was released more than 5 years ago, in October of 2003.

ChainBuilder (http://chainforge.net) is an open source integration engine that uses its own Enterprise Service Bus (ESB, http://en.wikipedia.org/wiki/Enterprise_service_bus). The ChainBuilder ESB components are Java Business Integration (JBI, http://en.wikipedia.org/wiki/Jbi) compliant, supporting common protocols and the parsing of healthcare standards like HL7 and X12. Interfaces are configured using plugins for the Eclipse integrated development environment (IDE), and it might take some previous development experience to get started.

XAware (http://xaware.org) is a Spring (http://en.wikipedia.org/wiki/Spring_Framework) based integration engine that was recently open sourced. Like ChainBuilder, XAware uses the Eclipse IDE for creating and deploying interfaces through plugins. Though XAware does not specifically target healthcare, it is a generic integration engine that supports many endpoints and protocols, including X12.

Mirth separates itself from other competitors and open source tools by its ease of use. Mirth can be downloaded, installed, and configured to have a custom interface running in a matter of minutes. After a few simple steps, any system can be configured to receive and/or send HL7 and many other healthcare standards. Although Mirth is open source, it is widely recognized to be feature comparable to commercial healthcare integration tools.

Mirth now has over 50,000 downloads and is seen by many as one of the top competitors in the healthcare integration engine space.

Conclusion

The combination of open source and a standards-based approach to healthcare interoperability has resonated with the global health information technology community. Healthcare executives welcome Mirth to their tooling portfolio because it is feature-comparable to commercial tools, accelerates interface development, and is professionally supported. Consequently, Mirth has been adopted by numerous hospitals, clinics, information exchanges, integrators, and application developers.

To improve Mirth, WebReach is now building additional and complementary open source infrastructure building blocks. For example, we are developing an open source Enterprise Master Patient Index (EMPI) that can be used to disambiguate patient identities across different healthcare applications, systems and organizations.

Jacob Brauer graduated cum laude from the University of California, Irvine in 2006 with a degree in Computer Science. He has been working for WebReach for 2.5 years, during which time he has been one of the lead engineers on the Mirth Project. Jacob has also contributed to other open source healthcare projects and initiatives, such as OpenMRS, an open source electronic medical record system framework designed for the developing world. He has partaken in many open source and healthcare conferences, including the Southern California Linux Expo Open Source Health Care Summit, Healthcare Information and Management Systems Society, Health Informatics Southern Africa, Open Source Health Care Alliance, and the HL7 Working Group Meeting.

"We expect to build the project ecosystem by encouraging the healthcare trading partners to participate as contributors, members of the project advisory council and early adaptors of tools delivered by the project."
http://www.openhealthtools.org/Reports/Jul08/Supply%20Chain%20Tooling.pdf

The Open Health Tools (OHT, http://www.openhealthtools.org) initiative is creating an ecosystem focused on the production of software tooling that promotes the exchange of medical information across political, geographic, cultural, product, and technology lines. At its core, OHT believes that the availability of high-quality tooling that interoperates will propel the industry forward, enabling organizations and vendors to build products and systems that effectively work together. This will “raise the interoperability bar” as a result of having tools that just work.

To achieve these lofty goals, careful consideration must be made to the constituencies that will be most affected by an OHT-influenced world. This document outlines a vision of OHT’s impact to these stakeholders. It does not explain the OHT process itself or how the OHT community operates. Instead, we place emphasis on the impact of that process within the health industry. The catchphrase “code is king” underpins this document, meaning that the manifestation of any open source community lies in the products and technology it produces.

OHT Stakeholders

To better understand OHT and the impact OHT can have within the industry, we will consider the interaction with three stakeholder groups that will be consuming or using OHT technology. It is important to highlight that each of the below categories focuses on individuals and not organizations.

Any given organization may have members in any or all of the categories below, either as employees or as beneficiaries/customers.

End-users: individuals that are directly interacting with systems that contain OHT code. These include caregivers, patients, administrators, case workers, and any other individual providing or receiving healthcare services or benefits using healthcare information technology (IT) applications.

Operational users: individuals that affect the purchase, deployment, operations, maintenance, and sustainment of healthcare IT systems and solutions within the organizations they support. This would include CIOs, system administrators, department managers, requirements managers, health informaticians, and IT staff.

Developers: individuals that design and produce working, executable implementations of software code that run on a machine. This includes developers, software engineers, product architects, and systems integrators.

From the above definitions, it is fairly clear that these communities are distinct and will have different expectations, interests, interactions, and ultimately different success measures vis-à-vis OHT. It is reasonable to anticipate that healthcare organizations will have presence across all categories and even some individuals are likely to span across groups. The next section will clarify these differences.

Stakeholders' Expectations

To better understand the stakeholders and their expectations, we discuss the following:

Current business challenges: to end-users, healthcare information is currently not available where, when, and in the format it is needed. There are inconsistencies and differences among systems with long wait times for IT staff to address these issues. For operational users, best-suite applications don't address all the business needs and best-of-breed applications create integration challenges and added expense. Inconsistencies in operational needs foster duplicative infrastructure, staffing, and support. For developers, significant investments are required to build any software infrastructure that is not core to their product offerings. Explosive heterogeneity further complicates development, especially for application programming interfaces (APIs) and standards support.

Operational objectives and expectations: end-users expect that systems should be reliable, easy-to-use, user-customized, flexible, and support new modalities such as mobile computing, cell-phones, and the Internet. In short, "IT shouldn't make my job any harder." Operational users expect systems to: i) be operationally reliable; ii) perform well; and iii) comply with industry drivers such as government mandates and security considerations. They also look for the ability to flexibly deploy, monitor, and manage installed systems. Developers expect software tools that provide flexibility and the ability to delivery high-quality code quickly. They also have a preference for tools that make tasks easier or which eliminate tedious chores and allow the developer to focus on the task at hand.

Anticipated benefit from OHT: end-users expect improved access to healthcare information and interoperability among vendor and supplier packages. Operational users expect a reduced integration burden, improved interoperability among off-the-shelf packages, and a reduced support burden resulting from more aligned industry offerings. Developers expect high-quality tools and components that "just work", improve productivity, and improve time to market delivery.

Point of intersection with OHT: the interaction with end-users will be indirect as they will use systems that have OHT components. Interaction with operational users will be direct as they will use tools and applications built using OHT technology. It will also be indirect as they specify purchases complying with industry standards implemented by OHT and as vendors are incented to engage in OHT. Developers will have direct interaction as they will use OHT tools, components, and other contributions to the OHT codebase.

Success measure: end-users expect reduced burden, improved job performance, and ubiquitous access to information where and when they need it. Operational users expect improved ability to effectively integrate products and interoperate within and outside of the business. OHT expects a controlled growth in the number of developers.

Healthcare Organizations and OHT

As a market sector, healthcare is particularly complex. These complexities are self-evident within healthcare organizations, as any given organization is likely to have most if not all of the stakeholders identified above. For OHT to add value to healthcare organizations, we must look beyond organizational labels.

To better understand the nature of the interactions of different organizations with OHT, and the business value that may be realized by participating, we provide an analysis of the touch points of different organizational roles with the OHT community.

Note that these organizational roles are categorized by function, such as providing care or paying for services, and not by the organizations themselves. Quite simply, this was done as many organizations take on multiple roles, especially when differences across countries are considered. We do not discuss a governmental role as we instead elected to enumerate the different types of activities government organizations may play.

Provider, payer, or public health organization: this role receives the following benefits from OHT involvement: i) the opportunity to incent and leverage co-investments with peer organizations; ii) improved de-facto interoperability among commercial applications; iii) improved ability to influence and impact commercial vendors; and iv) influence on OHT priorities. Contributions to the OHT community include use cases, subject matter experts, organizational requirements, development resources, funding, and code contributions. OHT will provide the following technology outputs: i) applications as demonstrated proof-of-concept or architectural prototypes; ii) tooling to facilitate custom integration or development; and iii) marketplace vendor product offerings aligned with requirements needs.

Oversight and regulatory: benefits include: i) reduced cost to test and assure implementations due to a shared codebase; and ii) improved marketplace compliance resulting from requirements support in an open source software (OSS) codebase. This role can contribute quality assurance, conformance, testing, subject matter experts, and requirements. It can also help to establish OHT priorities. Received technology outputs include the establishment of standard test harnesses and an OSS codebase conformant with oversight expectation.

Product vendor: benefits include: i) reduced cost of non-differentiating software product infrastructure investments; ii) increased market share resulting from overall community growth fostered by OSS; iii) improved quality resulting from a vetted OSS codebase; iv) improved sight lines to the needs of a potential customer base; and v) market branding and product positioning. Contributions can include code and development resources.

Integrator: benefits include: i) reduced integration risk resulting from enhanced vendor product interoperability; ii) improved sight lines to the needs of a potential customer base; iii) the ability to influence OHT priorities; and iv) market branding and positioning. Contributions can be resource contributions such as personnel or capital, subject matter experts, code, integration and implementation experience, testing, and quality assurance. Technology outputs include a leverageable codebase in the form of tools and applications.

Standards development organization: benefits include: i) reduced or eliminated custom tooling; ii) improved marketplace product support; iii) improved ability to develop standards; iv) improved value of healthcare standards produced; and v) the ability to influence OHT priorities. Contributions can include use cases, subject matter experts, organizational requirements, and funding. In return, the technology received includes tooling which supports the standards development process, healthcare standards-compliant market offerings, and an OSS codebase.

Foundation: benefits include tangible health community impact from investment and the ability to incent mainstream marketplace change aligned with the Foundation's objectives. Contributions include influence on OHT priorities and funding to the OHT community.

Technology benefits include a leverageable open source platform codebase, tooling, and interoperable commercial marketplace product support.

Conclusion

Spanning beyond cultural, organizational, and geopolitical lines, healthcare organizations the world over share a tremendous number of qualities and objectives. Objectives such as improving care quality and outcomes are shared needs. Fostering information quality, availability, and access is essential to success. Recognizing marketplace heterogeneity while still promoting interoperability is key. OHT fosters these objectives by providing an ecosystem where organizations across the domain can collaborate, engage, and ultimately produce solutions that work to realize these goals.

“Code is king” has tangible impacts and implementation matters. By establishing a common, open, available code infrastructure that can be used by organizations, vendors, and integrators, the bar to interoperability is raised and the costs and risks of doing so are reduced. OHT establishes an ecosystem into which a huge variety of constituents and organizations can effectively engage. This is of paramount importance, as it fosters the environment needed for collaboration, co-investment, and ultimately the development of solutions that address the diverse needs of the community while fostering the sharing of ideas and investment burdens.

Skip McGaughey is Executive Director of Open Health Tools. Open Heath Tools is a collaborative open source effort between national health agencies, major healthcare providers, researchers, academics, international standards bodies and companies from Australia, Canada, United States, United Kingdom and Europe. Its goal is to develop common healthcare IT products and services and provide software tools and components that accelerate the implementation of electronic health information interoperability. Skip was co-founder of Eclipse, a multi-language, multi-vendor open source platform for tool integration. There are over 800,000 organizations and four million developers using Eclipse. Eclipse pioneered the linkage between building open source software and enabling successful and profitable ecosystems to deliver technology to customers.

Ken Rubin is a senior healthcare architect with a leading systems integrator. His focus is informatics and electronic health records interoperability. He has supported the [US] Veterans Health Administration and the [UK] National Programme for IT. Mr. Rubin chairs committees for the OMG, HL7, Open Health Tools, and the Healthcare Services Specification Project (HSSP), and has been involved in healthcare for over a decade.

"The industry has learned by experience that the only software-related standards to fully achieve [their] goals are those which not only permit but encourage open source implementations. Open source implementations are a quality and honesty check for any open standard that might be implemented in software..."
http://www.opensource.org/osr-rationale

Many open source projects implement open standards. We interviewed five developers who implemented different open standards in open source projects to find out how much interplay there was between implementors and standards developers and how important this communication was as they programmed the details of the specifications. Our somewhat unexpected finding was that developers preferred to work from the printed specification, separate from the standards source. When asked for a reason, most reported that resource constraints prevented them from writing code and specifications at the same time; another factor was the satisfaction that comes from working independently. Most of the developers we spoke to were more than halfway through their development before they even considered reporting specification problems to the source organization. Although this speaks well for the overall quality of computer industry specifications, it also means that feedback from open source developers is not getting back to the specification's authors.

Open Source and Open Standards

Open source and open standards are not the same thing. Open source refers to software whose source code is freely available to users for reference, debugging, modification, and/or extension. Open standards are, typically, specifications: formal descriptions of software or software interfaces.

Open standards may have reference implementations, but the description in the formal standard typically takes precedence over the behaviour of a reference implementation.

It is interesting that the two phrases use the word “open” so differently: For open source, open means that the source code must be distributed with every copy of an executable application and every recipient must be allowed to modify and distribute the source code freely to subsequent users. In open standards, open signifies that the standards process is open to participation and that the completed standards are available to everyone. Working documents and drafts are typically kept private to the issuing organization’s members, and there may be reasonable conditions for participation such as membership fees, but any person or company may participate as a member at a meaningful level. Many standards organizations give copies of their standards away for free and the right to implement a standard is typically also free and, if not, is available on fair and equitable terms.

In computing, standards enable portability and interoperability. Portability includes: i) application code ports between operating systems or compilers; ii) middleware architecture ports between systems--its source code may also port but not nearly to the same extent; and iii) a developer’s skills porting from one platform to another. Each time something ports, someone saves time and money. Interoperability also pays, motivating companies to get together and write standards.

On examination, we see that the two approaches are complementary:

- open standards need implementations to provide: i) confirmation of their suitability; ii) a market presence; and iii) feedback from implementors and users

- open source development projects need guidance and direction regarding their interfaces for interoperability and portability

Each benefits from the products of the other and this synergy can be found in many projects.

We interviewed developers of open source tools based on open standards to investigate the interplay between open source development projects and open standards adoption and maintenance. We had expected that communication would flow liberally in both directions, but found that was not the case. Most of the development projects preferred to regard the specifications as read-only documents and were able to code productively straight from the printout. Our reports from five different projects and our conclusions follow.

ArgoUML

Jason Robbins started the ArgoUML project (http://argouml.tigris.org/) to build a modeling tool based on the Object Management Group's(OMG, http://www.omg.org/) standard unified modeling language (UML, http://en.wikipedia.org/wiki/Unified_Modeling_Language). He has since moved on to other projects, but stayed with this and a number of other standards-based OSS projects long enough to give us some good feedback. Jason considers this project to be a consumer of standards, and points out that a well-written and well-used standard will have thousands more readers and implementors than authors.

The usual case is a group isolated from the specification authors, coding along without a break unless they encounter an inconsistency or ambiguity of such consequence that they contact the issuing organization to have it resolved. Interpretation of words or phrases in OMG specifications is one of the largest sources of filed issues. Jason was the first person to point out something that we ultimately heard from many directions.

Any project that consumes specifications but which doesn't help write them must either accept the limitations of the standards as they are or delay implementing the affected part of the project until the problem is resolved by the issuing organization. Anyone may submit an issue about a specification to OMG using http://www.omg.org/technology/issuesform.htm, but it typically takes weeks or even months for these to be resolved. This is an eternity in open source development time. OMG has an accelerated resolution process that, for a restricted class of urgent issues, delivers a resolution in only two weeks. Rarely used, this process still lacks the quick response time required by most open source projects.

Marko Boger discovered the ArgoUML project after Jason Robbins had completed his work and moved on. It implemented only four diagrams and Marko needed another five for his work, so he gathered a group of students and started to code. As with our other example developers, the crew worked straight from the specification without contacting any of its authors.

One of the most interesting parts of the project was the building of a MOF-compliant (meta-object facility, http://en.wikipedia.org/wiki/Meta-Object_Facility) repository from the metamodel.

In a convincing test-case for code re-use based on standards, they convinced Novosoft to open up the source for its MOF-compliant repository toolxix, allowing them to update it to UML 1.3 and XMI 1.0 (http://nsuml.sourceforge.net/). Later they switched to MDR, a metadata repository from Sun's open source NetBeans project, enabling a move to UML 1.4 and XMI 1.2. This provided a sophisticated repository system, dynamically updatable from one version of the metamodel to the next, or to a specialized profile. Without an open standard for the repository structure and interfaces, the straightforward repository switch would have been difficult at best.

Partway through the work, they attended an academic conference where they met many of the authors of the specifications they were implementing. Direct contact continued after the conference, as the group stayed in touch as coding continued. Marko reports that this did in fact make a difference, but not in their coding to existing specifications. Instead, it made them more aware of parts of the specification they had overlooked and of new specifications under development.

Around this time, the ArgoUML team split: one group continued the open source development while the other, led by Marko, took a snapshot of the BSD-licensed source as the basis for a company which they named Gentleware. Using a conventional business model, they expanded ArgoUML into a set of modeling tools. They also joined OMG, submitted to the UML 2.0 diagram interchange specification, and chaired the committee finalizing the specification. This is an excellent example of the interplay between open source, open standards, and proprietary extensions.

AndroMDA

Matthias Bohlen's AndroMDA (http://www.andromda.org/) implements OMG's Model Driven Architecture (MDA), an ambitious achievement for an open source application. Matthias, a freelance consultant specializing in MDA, decided to write the first version of AndroMDA to show his customers its benefits.

OMG's MDA process starts with a Platform-Independent Model (PIM) of the target application's business functionality and behaviour, typically designed in UML. Two transformations carry this through to a coded application which is ready to make, deploy, and run. The first transformation enhances the PIM meta-objects with behaviour that enables the transformation to the meta-objects of the Platform-Specific Model (PSM). The second transformation takes the PSM meta-objects to code, build script, interface definitions, and whatever other artifacts are needed. AndroMDA uses an open source template engine for this task. The two steps are automated either partially or totally, depending on the application domain and situation. The AndroMDA user finally codes the real business logic mostly by hand, depending on the application domain and how much of the application logic calls for known patterns or uses standard UML profiles. In the process, the different steps, transformations, and code generation may be performed by one or more tools which are not dictated by the specification. XMI, OMG's standard format for model transfer, is used to transfer the various models between tools, enabling tool interoperability.

AndroMDA is rapidly becoming a very flexible and complete implementation of the MDA. A community of highly skilled developers and users has formed around the tool, and taken the product far beyond its first implementation.

Unlike ArgoUML and MICO (http://www.mico.org/), AndroMDA has benefited from communication with OMG during development, but not through its prime contributors. Matthias points out that they, like the other open source developers we've described, don't have time to develop code and standards simultaneously and must rely on others to carry their messages. Martin Matula of Sun Microsystems' NetBeans project and Marko Boger participate actively in OMG meetings. They inform the other contributors to the AndroMDA code base, who are not members of OMG, and who do not subscribe to any of the group's email lists.

AndroMDA doesn't mind reacting to changes in the specifications, a task Matthias described as "easy". He claims that it's more difficult for users to adjust than for builders. AndroMDA will use an abstraction layer to hide the differences between UML 1.X and 2.0 where they can, but users will have to adjust to new elements and capabilities, and will have to revise any cartridges they have written in order to use them with the upgraded tool version.

MICO

Arno Puder started MICO when he was a graduate student at the University of California at Berkeley. He originally wanted to develop a textbook and course that would teach coding of network middleware. Because design was not part of the process, he looked for a pre-defined system and found it in the Common Object Requesting Broker Architecture (CORBA, http://en.wikipedia.org/wiki/Corba) specifications. When he started, he was naïve about the ways of both open source and open standards. He obtained his copies of CORBA without studying their origin and named his project MICO for "Mini-CORBA". Keeping with the just-in-time philosophy, Arno brushed up on open source licensing as he prepared to post his early code on the web. After all, his goal was to write a textbook and the code was only the means to this end. After posting his CORBA implementation on the web, he found that it was widely used as middleware in software projects whose scope went far beyond the teaching tool originally envisioned. The MICO crew made it available under the original GPL, and then later under the LGPL at some users' requests, and found that its popularity expanded even further. The developers built it into a full implementation of the then-current CORBA specification, forcing a change of its name's significance from the no-longer-true "Mini-CORBA" to the recursive "MICO Is CORBA".

Arno and the other coders on the MICO project worked straight from the OMG documents. They were nearly finished development before they discovered how to submit issues to OMG through our website, meaning they only used the specifications for guidance. How well did they do, in spite of this isolation? Very well, as we'll see.

As the MICO crew was finishing its work, The Open Group (TOG, http://www.opengroup.org/) was developing a CORBA test suite under contract to OMG. The agreement called for three ORBs (http://en.wikipedia.org/wiki/Object_request_broker) to be certified and, since few vendors were ready for testing, OMG and TOG looked for open source ORBs. MICO was one of the lucky winners of a free certification, if it could pass the TOG tests. They ran the tests and, after resolving a few issues, qualified for certification. Arno traveled to the next OMG member meeting to receive his certificate and, for the first time, meet the people who wrote the specification he had implemented and find out how the technology adoption process worked.

This was after all of the major coding of MICO was complete, and far too late for the contact to help with coding decisions. Still, Arno doesn't feel that they needed much help–-after all, they completed an ORB that passed the compliance test without any contact. Arno continued to attend OMG meetings on behalf of Deutsche Telekom, his new employer, but his work centered on CORBA open source testing (COST) and other projects separate from the basic specifications coded into MICO.

Working at the University of Frankfort, Frank Pilhofer started on MICO by upgrading it to include the Portable Object Adapter (POA, http://en.wikipedia.org/wiki/Portable_Object_Adapter) and Objects by Value (OBV) which were both added to CORBA around 1998. Both are substantial and complex specifications with many interworking parts: POA defines the allocation and deallocation of resources as object instances are activated and deactivated, while OBV provides an object-like programming element.

Frank reports that he didn't encounter any serious issues as he implemented the POA and OBV specifications. He and the rest of the MICO crew used Internet sources such as the news group comp.object.corba to resolve questions they couldn't handle themselves, but preferred the independent feeling that came from handling most of these problems without asking outsiders.

Interoperability with other ORBs and language-mapping consistency provided more interesting moments than did pure implementation. CORBA is about interoperability, but open source projects don't have big budgets for software testing. They mainly tested against other free Open Source ORBs: TAO (http://www.cs.wustl.edu/~schmidt/TAO.html) and OmniORB (http://omniorb.sourceforge.net).

One incompatibility, encoding of valuetypes, was enough of a problem that they submitted it to OMG as an issue to be resolved by the CORBA Revision Task Force.

Issue resolution is a process that looks entirely different from the inside than from the outside, according to Frank who has worked this process from both ends. From the outside, all you want is a resolution so you can proceed. From the inside, you're faced with resolving an ambiguity in a specification that has already been built into commercial and open source products used by thousands of people. It's likely that a resolution will seriously affect these products, and representatives usually sit on the Task Force that is going to adopt the resolution. Seen in this light, the several weeks or months that it takes to resolve a set of issues takes on a much more reasonable look.

Frank and the MICO crew continued to submit issues as they worked, but stopped expecting resolution to come quickly enough to be built into their current code. Instead, they took stopgap measures to get by, realizing that they'd have to make a suitable change when the issues were eventually resolved.

CORBA Component Model

As Frank completed his work, Arno arranged for funding for a related coding project, the CORBA Component Model (CCM, http://en.wikipedia.org/wiki/CORBA_Component_Model#CORBA_Component_Model). Originally written by several major system vendors, the CCM needed support in its finalization phase in order to complete the full course of adoption and enter the OMG specification book. Unfortunately, the market had focused almost exclusively on Enterprise JavaBeans (EJBs) and the multi-language CCM superset with its additional features was being ignored.

Alcatel was willing to fund Frank's work and he found a few willing helpers when he showed up at OMG meetings ready to work. Alcatel funding covered Frank's coding time and his attendance at meetings. They expected him to feed his experience back and see that a finalized CCM was adopted at the end of the process.

This is our target case: an open source project done by a member of a specifications consortium, in full communication with its other members. However, it lacks many components that make open source special, such as the open organization with its large, widely spread body of voluntary contributors. Frank's MICO CCM implementation is available as open source under the GPL/LGPL, but it was written by a single developer hired by an enterprise to do a particular project. The availability of code as open source was incidental to the rest of the project, which would have run identically if the code had been kept proprietary. The resulting CCM specification benefits from an open source developer, but not from the community of contributors typically associated with the phrase.

Conclusions

Specifications flow from consortia, providing direction for many open source projects, proprietary products, and inhouse software development. Because the specifications are downloaded anonymously, no reverse communication channel opens from specification user to supplier. The sources we interviewed for this paper all agreed that communication along this channel will be sparse because developers are too busy to produce code and comment on specifications at the same time. We don't agree that this is the most productive way for open source projects and open standards organizations to work together.

As sophisticated and experienced implementors, open source developers clearly have much to contribute to the design and specification of the applications they ultimately produce, and the industry will benefit from their early and deep involvement in the standards process. We also believe that standards organizations need to recognize and account for open source implementations, a task which requires input from the open source community. Recent cooperation between OMG and Eclipse shows that this can work in a formal way and we look forward to more examples in the future.

Open source projects produce another output besides feedback: well-written applications that faithfully implement the specifications on which they are based. It says something good about a standard, and in turn about the organization that produced it, every time an outside project takes a specification and turns it into an application. Consortia should listen to this message and be pleased.

This article is a shortened version of the OMG publication "Open Source and Open Standards: -Working Together for Effective Software Development and Distribution". The full version is available from the OMG website at http://www.omg.org/docs/omg/04-10-02.rtf.

Dr. Jon Siegel is Vice President of Technology Transfer for the Object Management Group. He holds a doctoral degree in Theoretical Physical Chemistry from Boston University.

Dr. Richard Mark Soley is Chairman and Chief Executive Officer of the Object Management Group. Dr. Soley holds the bachelor's, master's and doctoral degrees in Computer Science and Engineering from the Massachusetts Institute of Technology.

What Factors Lead to Innovation Failures? Evidence from a Large Dataset

Copyright: Rejean Landry, Nabil Amara, Nizar Becheikh

From the Abstract:

This paper argues that a better understanding of innovation failures would contribute to improve our understanding of innovation successes and that it would carry practical implications for both the research and management of the innovation process. This issue is addressed by asking three questions: What are the factors that lead innovation projects to termination? What is the impact of the degree of novelty of product innovation on the termination of innovation projects? Are there differences in the factors explaining the innovation projects that succeed and those that are terminated by managers?

http://kuuc.chair.ulaval.ca/fichier.php/64/WP-2008-02-Landry+Amara+Becheikh-IAMOT.pdf

The Tower and the Cloud

Copyright: Educause

From the Description:

The emergence of the networked information economy is unleashing two powerful forces. On one hand, easy access to high-speed networks is empowering individuals. People can now discover and consume information resources and services globally from their homes. Further, new social computing approaches are inviting people to share in the creation and edification of information on the Internet. Empowerment of the individual—or consumerization—is reducing the individual's reliance on traditional brick-and-mortar institutions in favor of new and emerging virtual ones. Second, ubiquitous access to high-speed networks along with network standards, open standards and content, and techniques for virtualizing hardware, software, and services is making it possible to leverage scale economies in unprecedented ways. What appears to be emerging is industrial-scale computing—a standardized infrastructure for delivering computing power, network bandwidth, data storage and protection, and services. Consumerization and industrialization beg the question "Is this the end of the middle?"; that is, what will be the role of "enterprise" IT in the future? Indeed, the bigger question is what will become of all of our intermediating institutions? This volume examines the impact of IT on higher education and on the IT organization in higher education.

http://www.educause.edu/thetowerandthecloud/133998

November 19

Reducing the Cost of BI Ownership

Toronto, ON

At this free breakfast seminar, learn about Open Source BI software and view live demos of zero-cost Business Intelligence tools.

http://www.sqlpower.ca/page/breakfreeseminar

November 25

Eclipse DemoCamp

Vancouver, BC

The Eclipse DemoCamps are an opportunity to showcase all of the cool interesting technology being built by the Eclipse community. They are also an opportunity for you to meet Eclipse enthusiasts in your city. You don't need to be a software developer to attend.

http://wiki.eclipse.org/Eclipse_DemoCamps_November_2008/Vancouver

November 26

YOW10 - C/C++ Programming

Ottawa, ON

This free evening event is dedicated to C/C++ programming. It will cover makefiles, shared libraries, subversion, and Eclipse.

http://www.osbootcamp.org/index.php?page=yow10

November 27

Eclipse DemoCamp

Ottawa, ON

The Eclipse DemoCamps are an opportunity to showcase all of the cool interesting technology being built by the Eclipse community. They are also an opportunity for you to meet Eclipse enthusiasts in your city. You don't need to be a software developer to attend.

http://wiki.eclipse.org/Eclipse_DemoCamps_November_2008/Ottawa

October 20

Calling Canada's Librarians - the Canadian Public Domain Needs You!

Toronto, ON

Access Copyright (The Canadian Copyright Licensing Agency) and Creative Commons Canada, in partnership with Creative Commons Corp. and the Wikimedia Foundation, invite Canada's library community to help us test the Canadian Public Domain Registry beta website. The ground-breaking project – the most comprehensive of its kind in Canada – will create an online, globally searchable catalogue of published Canadian literary works. The Registry's integrated rights calculator allows users to automatically determine each work's copyright status on an evolving basis. The Registry will also link to digital versions of the work and provide information about where a paper-copy can be purchased, when available.

http://creativecommons.ca/blog/?p=277

October 30

LAC Launches Flickr/YouTube Project

Ottawa, ON

Library and Archives Canada is pleased to announce that in anticipation of the 2008 Irish Studies Symposium, a selection of digital images related to Irish-Canadian documentary heritage are now available on Flickr.com, a popular photo-sharing community. A selection of video presentations from the upcoming Symposium will also be added to YouTube.com in November 2008. Visitors to the Library and Archives Canada's album at Flickr.com are encouraged to explore the interactive image collection which allows for commenting and tagging of content. All images on Flickr.com are tagged with geographic information to allow visitors to explore history in the context of their surroundings by navigating the album on a virtual map of the world.

http://www.collectionscanada.gc.ca/whats-new/013-359-e.html

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

December 2008	Enabling Innovation Guest Editor: Steven Muegge
January 2009	Enterprise Participation Guest Editor: Donald Smith
February 2009:	Commercialisation Guest Editor: Robert Withrow
March 2009:	Geospatial Guest Editor: Dave McIlhagga
April 2009:	Open APIs Guest Editor: Michael Weiss
May 2009:	Open Source in Government Guest Editor: James Bowen

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.

Open Health Tools

National Health Services, US Government Agencies, Providers

Canada Health Infoway
National Health Service (NHS, UK)
National eHealth Transition Authority (NEHTA, Australia)
Veterans Health Administration (VHA, US)
Mayo Clinic

Standards Organisations and Open Source Foundations

Eclipse
Health Level 7 (HL7)
Health Services Specification Project (HSSP)
International Health Terminology Standards Development Organisation (IHTSDO)
Object Management Group

Research and Academic Organisations

Australian eHealth Research Centre (AEHRC)
Linkoping University
Mohawk College
Oregon State University

Companies that Supply Healthcare Products and Services

BT Health
B2 International
Cambio Healthcare Systems
Collabnet
Collaborative Software Initiative
Discovery Machine
IBM
Innoopract
Inpriva
International Technology Group
JP Systems, Inc
Kestral
Misys
NexJ
Ocean Informatics
Oracle
Ozmosis Inc
Palamida
Red Hat
Friends of eHealth

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.