The Case For An Open Source Shared Earth Model

A white paper by

R.T. Rundle, R Vuurboom
OpenSharedEarth.org

The software industry as a whole has been transformed by the rise of open source software. In the beginning the impact of open source was thought to be unimportant or even damaging to the real business of commercial closed-source software. There simply had to be something intrinsically wrong with the idea of intellectual property simply being given away. Free intellectual property had to of poor quality or else if it was high quality it had to be stolen. This was the conventional wisdom.

Yet, with regard to open source software, the conventional wisdom has been proved irrefutably wrong. There is now a huge volume of high quality open source software widely available to be repurposed for any use. Counter intuitively free intellectual property is not inconsistent with for-profit enterprises. Red Hat Inc., a software company whose foundation is the open source Linux operating system, has a $15 billion market cap.

To be certain, the acceptance of open source software has been uneven among various industries. The finance and education industries have heavily embraced open source software. It is also an unstated fact that the entire Internet has been built on open source. It is not clear why certain industries have embraced open source while others have largely spurned it. Perhaps the most likely explanation of why open source has succeeded in some places and not others is that the entire software industry as a whole is very young. If one accepts this hypothesis, then over time one can expect an increasing penetration of open source in all industries.

The oil and gas industry in general and shared earth modeling in particular are places where very little open source influence is felt. This white paper examines the tremendous impact that a successful open source initiative might have in this industry. It ends with a call to action to create an open source consortium to promote and manage a publicly available open source repository specifically for shared earth modeling.

Knocking Down the Shibboleths

First we need to address a number of ideas about open source that have currency only in the industries, such as oil and gas, which do not have a history of support for open source.

The most pervasive and pernicious of these ideas is that one that open source represents some kind of creeping socialism or even communism promoted by a community of hippie programmers who want nothing less than to destroy the entire commercial software industry. Indeed there are some, like Richard Stallman aka “The Last Hacker”, who fit this mold exactly. [1] Stallman believes that software should be free like air. In Stallman’s world programmers make a living in much the same way as a physician makes a living. Using Stallman’s logic, medical knowledge is free and the physician’s fee is based on their understanding of this knowledge. In the same way a programmer should live off services rendered through the understanding of an extensive body of free software. Even though Stallman has had a great influence on the open source community especially through his ideas about software licensing, it is important to understand that his view is a minority view.

The second idea is that open source is inherently low quality. This belief seems to flow from the bromide that "you get what you pay for" and therefore since open source is free it must be shoddy. After all if programmers are giving away their labor it must mean they are unsuitable for gainful employment making software someone will pay for. The "open source means low quality" idea is easily dispelled with a survey of all the high quality software that is open source…a list too long to be even sampled here. It is not an exaggeration to say that the entire foundation of the Internet is open source. Also to assume that open source software is only produced by unpaid programmers is to completely misunderstand the economics of open source (more on this later).

Finally it is assumed that open source must be insecure. After all if a software product is an open kimono then it would seem that attackers could easily ferret out flaws and weakness to be used to suborn it. This turns out to be the most naïve misunderstanding of open source of all. Indeed flaws and weaknesses of open source software are open to examination by all. This allows a vibrant community of security researchers to find and document problems so that they can be promptly addressed. It is an article of faith that security holes in all software, whether open or closed, will be found and exploited and therefore by allowing a broad-based and public examination of open source, security is inherently better. As has been proven over and over, obscurity is not security. Perhaps the most important secure protocol, OpenSSL, which is used by millions of people to conduct sensitive online transactions involving bank accounts and credit cards is completely open source.

Thus let us begin an in-depth discussion of open source in the oil & gas industry with the basic understanding that open source is suitable for commercial development, is capable of being high quality and is capable of being secure.

Proprietary Software vs. Open Source

It is difficult to have a balanced discussion of the merits of open source versus proprietary software as the arguments for each side tend to take on a religious fervor that brooks no dissent. Each side seems to believe only in total victory. In truth there is a place for both propriety software and open source software in any industry and especially in the oil & gas business.

The case for proprietary software is straightforward. The source for a software product represents nothing less than the accumulated intellectual property of a company whether developed internally or acquired through purchase. Intellectual property more so than other types of property is vulnerable to theft and misuse. Software can be stolen without the owner's knowledge. Stolen software can then be used to create a competing product which unfairly siphons off revenue from the original. In such situations legal recourse is difficult or impossible because the software will have been modified in subtle ways so as to be unrecognizable when compared against the original. Therefore to protect the investors of the company, the source code for the product is maintained as a trade secret and worked on only by a carefully vetted team of employees and contractors. To do otherwise would be fiscally irresponsible.

A part of the revenue from the sale of the software product is, when managed responsibly, reinvested in the enhancement and maintenance of the software product. In this way a successful software product evolves and continues to win greater market share over time.

The rationale behind proprietary software varies little from the viewpoint of a young Bill Gates, who in a famous February 3, 1976 letter complained that most of the users of his Altair BASIC program had stolen the software rather than purchased it:

> "As the majority of hobbyists must be aware, most of you steal your software. Hardware is paid for, but software is something to share. Who cares if the people who worked on it get paid?" [2]

Protected proprietary software does not begin or end with the source code of course. The resulting product (often called the binary) must also be locked down. This is done through increasingly sophisticated hardware and software mechanisms such as licensing servers and hardware dongles. People who worked on the software must be legally bound through non-compete and non-disclosure contracts from impinging the reputation or the exclusivity of the company's intellectual property.

The case for open source software is much harder to make. It our view it does not begin with a purist's view that software "should be free like air." It is not a "new form of communism" as Bill Gates was once quoted as saying. Our view of open source is one that is compatible with commercial software development and one that can coexist and even be comingled with proprietary software. How is this possible? To quote again from Bill Gates's letter: "What hobbyist can put 3-man years into programming, finding all the bugs, documenting his product and distribute [it] for free?"

However money can be made from open source. There are in fact commercial enterprises based on open source. Red Hat, Inc., already mentioned is one, Rack Space which is based on the open source Open Stack cloud computing software and Cloudera which is based on Hadoop and a few others. The economic model in each of these companies is the same: the core intellectual property is freely available while each company's economic interests are served by providing support, training, documentation, and in many cases a metered ecosystem in which paying subscribers can implement solutions.

It must be unequivocally stated that the open source business model is much more difficult to implement and there are very few successful open source commercial enterprises while there are many, many successful proprietary software companies. As venture capitalist Peter Levine states in a blog post

> "[T]he more successful an open source project, the more large companies want to co-opt the code base. I experienced this first-hand as CEO at XenSource, where every major software and hardware company leveraged our code base with nearly zero revenue coming back to us. We had made the product so easy to use and so important, that we had out-engineered ourselves. Great for the open source community, not so great for us." [3]

So vendor economics is not one of the compelling rationales for open source. What is happening in an open source software company is that it is selling a commodity rather than a differentiated product. Vendor competition in a commodity space is a vicious race to the bottom: the most efficient, the lowest cost producer will win. Axiomatically there is little revenue that can be re-invested in the product.

But what about consumer economics? Again the case for proprietary software is straightforward. The consumer pays for a license and for support. There is a great incentive for the vendor to provide high quality software and support so that a satisfied user base will lead to ever expanding license sales. The consumer must endure the various intellectual property protection mechanisms the vendor has imposed which greatly limits the ways that the software can be used. The consumer must anticipate the type of problems to be solved and the size of the problems to be solved in order to purchase the right number of licenses, the right amount of support and the right options for the software. These are often difficult decisions, but the costs are obvious and can be tabulated directly from the vendor's price list.

Open source economics from a consumer standpoint is much more problematic. Free software is not free. Indeed the software can be distributed to an unlimited number of users and used to solve problems in unrestricted ways, however the burden of ensuring the effectiveness of the software has moved from the vendor (i.e. the open source community) to the consumer. For the vast majority of consumers this is a very unwanted burden entailing a lot of risk and hidden costs. In going down the open source route the consumer will often contract outside help or purchase support services from the vendors who have chosen to create the open source business model described above. Compared to proprietary software, the costs of implementing open source software cannot be easily calculated.

If the neither the vendor economics nor the consumer economics are compelling rationales for open source, what is the compelling case for open source?

The Consortium Model for Open Source

As it turns out we believe the most compelling case for open source, especially in the oil & gas business, is when both vendors and consumers participate in a consortium. That is both vendors and consumers contribute software to the consortium. The software in the consortium becomes shared intellectual property which can be leveraged by both vendors and consumers. The vendors leverage the common intellectual property to enhance their proprietary offerings. The consumers leverage the common intellectual property to build custom solutions that are otherwise unavailable in the marketplace. Everyone wins.

Vendors and consumers pay a recurrent fee to participate in the consortium and are encouraged to donate software and personnel to shape the offerings of the consortium and otherwise support the common intellectual property. Vendors have an incentive to join because their proprietary offering will more likely to interoperate with other vendors. Consumers have an incentive to join because they will be more likely to implement solutions to their problems. The consortium is not a threat to vendors because only the most basic, commoditized components will become part of the common intellectual property. The consortium on the other hand will be a boon to consumers because they will no longer pay outrageous prices for file converters and other low tech solutions. In the end, the consortium, if it achieves critical mass, will make oil & gas software industry a much more rational marketplace where high prices are paid for innovative and research-intensive solutions and solutions to common problems are essentially free.

Consortia of this type already exist of course in the oil & gas industry in the form of various standards organizations. Energistics is a great example of such an organization. [4] The membership of Energistics represents a Who's Who of the energy business, from the majors and super-majors down to one person consulting firms. The membership fee of Energistics is a sliding scale based on the size of the member organization which permits (1) the organization to be adequately funded and (2) technical participation of even the smallest members on an equal footing with the largest.

Even more intriguing is that Energistics is creating software standards. These standards are implemented in the form of XML documents which can be uses as a basis for software solutions. Its WITSML drilling protocol is well designed and is broadly accepted. A large number of vendors and consumers support the standard and a large number of products have been built that support it.

If a consortium already exists that produces software standards, why then, it must be asked, is another consortium producing open source software needed? There are basically 2 reasons: (1) the scale of the problem and (2) the speed of the solution. We elaborate on these reasons in the next 2 sections.

The Scale of the Problem

The areas of standardization that Energistics addresses are quite small relative to the need for standardization. Energistics addresses well drilling data exchange (WITSML), well production data exchange (PRODML) and earth data modeling (RESQML). These cover a number of key subsurface objects including wells, wellbores, grids, reservoir properties and so on. However there are even more key subsurface objects that are not covered including PVT models, seismic data cubes, well completions, well casing design and many, many other objects critical to a comprehensive understanding of the subsurface.

Perhaps even more problematic is that these standards have been built largely for a fixed constituency which is incompatible with other constituencies. For example WITSML is a drilling standard. As such it not particularly suitable for seismic interpretation or reservoir modeling. The WITSML well trajectory is a set of control points which precisely define the curvature of the wellbore in 3-space for the entire length of the well. Seismic interpretation and reservoir modeling need only a simple point-to-point description of the wellbore in the reservoir only. Naturally it is a straightforward process to simplify the WITSML well trajectory into a list of points on the wellbore, but this simply one of many adaptions that must be made in software using the standard. The net result is that WITSML will always exist at the periphery of a delivered product and never at the core. There must always be a mapping to and from the WITSML data object and the native data object that each software product implements. This mapping requires considerable effort to achieve and is a source of many deficiencies and misuse or misunderstanding of the standard.

Indeed it will be pointed out that a place at the periphery was all that WITSML aspires to be. By definition it is an exchange format, designed to be a kind of glue between competing vendors. One of the fundamental questions this paper asks is, is this good enough? Is the tremendous effort being put into the standards really delivering value? The answer is ambiguous at best.

The problem with implementation of the WITSML standard is that each vendor interprets the standard differently. Worse some vendors feel that they have sufficient market share that they can simply get away with willfully ignoring parts of the standard. As might be expected, either willful or accidental misapplications of the standard are disastrous to inoperability which is, in fact, a core aspiration of WITSML. The lack of inoperability was exceeding bad with version 1.3 of the standard where vendors were allowed to “self-certify”, a kind of honor system which was as inadequate as one might expect. In 1.4, the latest version of the standard, vendors are no longer allowed to self-certify but must subject their products to a standardized suite of tests to verify compatibility with the standard. This results in much better inoperability but is still inadequate. Vendors can still decide which parts of the standard to implement. As of this writing, the WITSML 1.4 compatibility test results for the Halliburton WITSML server show that of 75 compatibility tests, 33 were "skipped". [5] Yet the Halliburton WITSML server is permitted to carry the WITSML 1.4 compatible imprimatur.

As a result of inadequate interoperability, products from competing vendors cannot be connected without rigorous testing to verify that they, in fact, interoperate. In practice what happens is that the products will work at a basic level but will have various subtle problems. Subtle problems cannot be ignored because often these types of problems involve engineering units which create large errors in the data exchange. The best way to insure proper operation of the WITSML standard is to choose the same vendor for each end of the exchange. Any 3rd party product implementing one end of the standard must rigorously test their implementation with each vendor's product that might be encountered on the other end and then adjust their implementation so that each of the various "flavors" of the WITSML protocol are supported. One might reasonably ask what benefit a standard imparts if one cannot count on its faithful and repeatable implementation. The vendors themselves would be hard pressed to answer this question. The final insult to the WITSML standard is that the standard it was intended to replace, WITS, is still going strong and is unlikely to be abandoned any time soon.

Compare and contrast the performance of the WITSML standard with standards from other industries. Take for example the Wi-Fi standard, a product of the Wi-Fi Alliance, a consortium of nearly 700 companies world-wide. It is a given that one can go down to the local big box store, buy armloads of Wi-Fi devices from a dozen competing vendors, go home and hook it all up and everything will work perfectly.

Why then is WITSML so inadequate? The point being made in this section is not that WITSML has been so poorly conceived, it is that the scale of the problem is so big. In stark contrast to the industry served by WiFi, the oil & gas industry is dominated by a small number of large players who have little incentive to play nicely with each other.

The reason the oil and gas industry has struggled with standardization is best captured in an unpublished paper by David Cotton, et al:

> *The oil and gas industry can aptly be labeled a latecomer industry, meaning that it has been slow to adopt digital technologies and standards. The oil and gas industry is not completely devoid of digitization or standards, but rather it has been more local and segmented than many other industries.*
>
> *A common trait of latecomer industries is that firms have specialized, complex industrial machinery involving large capital investments. The large capital costs lead to cyclical product demand as well as a barrier to entry for other firms. This has the effect of making it difficult for standards to gain momentum, especially when profits are high. [6]*

WITSML is not being held up here as a failure. It is, in fact, despite all its many flaws and limitations, a resounding success with broad uptake throughout the industry. Compared to the various other standards, it is arguably the best information technology ever developed for this industry. As far as interoperability and technical excellence, it is head and shoulders above other widely accepted standards such as SEG Y, DLIS and LAS. The point being made is that the best is simply not good enough. It is simply inadequate to the scale of the problem.

The Speed of the Solution

No proof need be presented that standardization processes are slow and tedious. Even under the best circumstances a new standard might take a decade from conception to implementation. Despite the time and resources involved, there is no guarantee of success. There seems to be only one way to succeed in a standardization effort and many ways to fail. In a fascinating 2011 paper entitled "Why Standardization Efforts Fail", Carl Cargill, an expert on software standardization, proposes that "no one really knows" how standards succeed. He goes on to detail 6 major categories of failures of standards with examples. Then there is this *cri de coeur* in the midst of his paper:

> *This then brings up the question, who creates standards? Basically, non-rational human beings create standards. I use this phrase advisedly, since most economic behavior (and hence, studies of standards) tends to assume a rational economic model. I have watched too many standardization efforts become a complex contest between corporate wills and a need to maintain a facade of control to believe that rational economic decisions are made. It should be remembered that a participant in a standardization effort wears many different hats simultaneously—hats that cover professional pride (doing what's right), corporate or organizational goals (doing what's right for your company), standardization organizational goals (doing what's right for the organization and in scope or charter and following the rules), a national interest (doing what is right for your country's industrial, social, or legal policies), and personal friendships (doing what's right to make you feel good and for social and professional strokes). When you put two dozen people with conflicting emotions, goals, backgrounds, and personal motivation in a room, ask them to decide on a complex interface whose future characteristics may or may not impact the market, and then provide minimal guidance and no enforceable deadline, one is hard pressed to describe the outcome as a rational economic decision. When you toss in a rapidly changing external environment, competing organizations doing the same thing, and a generalized need to cooperate rather than compete, you have the basis of some interesting decisions that create standards.* [7]

To anyone who has been involved in any type of standardization process, this screed rings so true it is spine tingling.

Rather than review the reasons the standardization process is slow and fraught with peril, it might be instructive to look at a standardization organization that works extremely well, is highly efficient and moves extremely fast. That organization is the Internet Engineering Task Force (IETF), the organization behind the standards that govern the Internet. The IETF has to be alone among standards organizations in producing eminently readable standards documents, the beloved RFCs or Request for Comments. Not all RFCs are standards but should an RFC become a standard, there is no question about its authority, its effectiveness or the subsequent uptake among affected parties. A IETF standard can move from draft to proposed to full standard in as little as 10 months, light speed in standardization terms. We sometimes talk about operating on "Internet time". It seems that the standards organization that governs the Internet does just that.

Yet, the IETF, under casual examination, would not appear to be an effective organization at all. The IETF is not a corporation, has no board of directors, no members and no dues. Standards are produced by Working Groups that anyone can join or quit at will. Standard acceptance cannot be voted on, since no one knows how many people are members of the Working Group and hence there is no concept of quorum or majority. Consensus on proposals is determined by "humming". According to RFC 4677, "The Tao of IETF", "if you agree with a proposal, you hum when prompted by the chair; if you disagree, you keep your silence." [8] Such a chaotic organization with such naïve processes could not possibly be effective…except that it is.

The success of the IETF is even more remarkable when it is remembered that there was a competing set of standards, the Open Systems Interconnection or OSI, whose charter was nothing less than replacing the whole of the Internet as we know it today. The OSI had the backing of governments, academia and the entire telecommunications industry. The organization behind OSI spent 10 years and an enormous amount of resources producing a series of very well regarded standards using a very disciplined and mature set of processes. The end result was abject failure. It is impossible to pinpoint the exact moment of failure but a turning point must have been when the OSI advocate Brian Carpenter gave a talk entitled "Is OSI Too Late?" at a meeting in 1989 and received a standing ovation. [9] It seems that the merry band of pranksters behind the IETF had won. The secret of success of the IETF seems to be embodied in an early quote about the IETF from David Clark: "We reject kings, presidents and voting. We believe in rough consensus and running code."

The point of this section is to say that whereas there is much to respect about disciplined and mature standards-making processes and organizations, there are simply some industries and some problem domains where this approach is simply too slow. Such an industry is the oil & gas industry and such a problem domain is subsurface modeling. The technical basis for oil and gas exploration and production is moving so fast that traditional standards-making cannot hope to keep up. As one example consider unconventional well technology. In 2006 few people in the industry knew what hydraulic fracturing was. In 2016 unconventional well technology dominates. This is simply the latest of various technologic changes that have swept the industry, previous examples being 3D seismic and horizontal drilling. Standards made the old fashioned way cannot possibly hope to keep up.

What is needed in subsurface modeling is a set of processes and an organization that is less like OSI and more like IETF.

The Open Shared Earth Consortium

The authors propose a consortium as a repository for running code to which all members of the consortium have access. Members are free to exploit this code for commercial purposes or internal use without incurring additional license fees and without suffering restrictive licensing schemes. Members pay dues for membership and donate code and labor to the consortium. Members can be any type of organization whether oil company, service company, software publisher or independent consultant. Membership dues are scaled to the size of the member organization and its desire to participate. The influence of the member on the consortium is also scaled in the same way.

The overarching purpose of the consortium is not to produce standards but to produce running code. Running code, when widely distributed and freely available, becomes a de facto standard. This code will have a type of open source license which is amenable to commercial use and will not in any way taint proprietary code with which it is mixed.

The scope of the platform offering of the consortium is nothing less than the entire problem domain as suggested by the needs of subsurface modeling. At a minimum this scope includes seismic interpretation, gridding, property modeling, reservoir simulation and a lot more. What is envisioned is a complete seismic processing to reservoir simulation workflow.

The purpose of the consortium is not to create technical innovations but to capture existing and mature technical innovations into a software platform that can used as a launching pad for member organizations to create new and proprietary innovations.

The business model of the consortium is not a threat to any organization, including software publishers. The consortium will be threat to software publishers only to the extent that software publishers insist on demanding extraordinary prices for mundane technology.

Such an ambitious mission statement prompts a host of questions. How can donated code be molded into a usable, extensible platform when this code will come in a veritable Tower of Babel of computer languages: C, C++, Fortran, Java, JavaScript and many others? How can a comprehensive solution to the subsurface modeling problem be fashioned from a dog's breakfast of donated software? Given such a huge scope, how can confidence in the consortium be maintained in an industry where software organizations of all types promise much and deliver little?

There are straightforward and compelling answers to all of these questions and more. However, these answers are beyond the scope of this white paper whose intent is limited to making the case for open source software in the oil and gas industry. Software engineering processes, product architecture and product deployment will be described in a succession of white papers to come.

What should be obvious however is a tremendous amount of effort is required to make this idea of a foundational set of shared earth software a reality. At this moment we are at the very beginning of the arc of this idea and so far there is no reason to assume that it will succeed given the path is strewn with failures of a similar nature. However the reason for starting down this path is that, should this effort succeed, the upside for member organizations would be huge. Take a moment to imagine an environment in which simple problems have simple solutions. Imagine being able to focus engineering effort and increasingly scarce capital on new and unsolved problems instead of squandering it on problems that have been solved over and over again. Imagine a frictionless and secure transfer of technology and data between and within organizations. Imagine a scalable collaboration of industry experts where those experts are focused entirely on the problem at hand and not on overcoming the various incompatibilities of their tool sets. Not only is such an environment possible, the authors believe it within reach.

The name of the consortium is OpenSharedEarth.org. If you or your organization would like participate in this effort please contact us through the website.

[1] J. Markoff, "One Man's Fight for Free Software," *New York Times,* p. 2, 11 January 1989.

[2] "Wikipedia," Wikimedia, 3 February 1976. [Online]. Available: https://en.wikipedia.org/wiki/Open_Letter_to_Hobbyists. [Accessed 24 July 2016].

[3] P. Levine, 13 February 2014. [Online]. Available: https://techcrunch.com/2014/02/13/please-dont-tell-me-you-want-to-be-the-next-red-hat/. [Accessed 24 July 2016].

[4] "Energistics.org," [Online]. Available: http://http://www.energistics.org. [Accessed 24 July 2016].

[5] "Haliburton WITSML result," Energistics, [Online]. Available: http://www.energistics.org/Assets/halliburtonwitsmlresult.txt. [Accessed 14 July 2016].

[6] D. Cotton, M. Grissonm, D. Spalding and W. R., "Standardization Barriers in the Petroleum Industry," 4 May 2012. [Online]. Available: http://morse.colorado.edu/~tlen5710/12s/Petroleum.pdf. [Accessed 24 July 2016].

[7] C. F. Cargill, "Why standardization efforts fail," *Journal of Electronic Publishing,* vol. 14, no. 1, 2011.

[8] P. Hoffman and S. Harris, "The Tao of IETF: A Novice's Guide to the Internet Engineering Task Force," September 2006. [Online]. Available: https://www.rfc-editor.org/rfc/rfc4677.txt. [Accessed 24 July 2016].

[9] A. L. Russell, "OSI: THe Internet That Wasn't," ieee.org, 30 July 2013. [Online]. Available: http://spectrum.ieee.org/computing/networks/osi-the-internet-that-wasnt. [Accessed 13 September 2016].