GESMER UPDEGROVE LLP
Attorneys at Law

Consortium Standards Bulletin

*A **ConsortiumInfo.org** publication*

MAY 2006
Vol V, No. 5

FEATURE ARTICLE

THE FREE STANDARDS GROUP: SQUARING THE OPEN SOURCE/OPEN STANDARDS CIRCLE

Andrew Updegrove

Abstract: *The creation of open standards for software and the development of open source software seem as different as night and day, due to the great differences between the two end products: the former describe features of, and interfaces between, software, while the latter is the software itself. Again, the utility of the former relies upon mandating that corresponding elements of compliant products remain unchanged, while a feature of the latter (guaranteed by licensing terms) is that it may evolve on a constant basis. But achieving the full value of open standards and open source software in the modern information technology world requires that open source software support open standards, and therefore that these very different tools be developed in a coordinated fashion – a seemingly insoluble dilemma. The following interview with the Free Standards Group, a voluntary, consensus-based non-profit organization formed to create standards for Linux and other key software in the open source development "stack," demonstrates how this difficult goal can be achieved, based upon novel techniques and the agreement of both constituencies to coordinate their activities in a real-time, collaborative fashion to the mutual benefit of those involved, as well as to the end-user community.*

Introduction: Open source software (OSS) and open standards share tantalizing similarities and frustrating differences. The similarities include the promise (for end users) of a greater range of less expensive products, and (for vendors) reduced research and development costs, lower risks, and larger and faster-emerging markets. Unfortunately, the differences include historically incompatible licensing regimes and profoundly different attitudes towards the dynamism of software.

During the early years of the open source revolution, these differences caused few problems. This was because the first OSS projects created discrete, stand alone programs, and few software standards had historically been encumbered by licensing terms that were incompatible with open source licenses. In recent years both of these situations have changed, increasing numbers of OSS products have been developed, and more and more patents have been asserted against key standards as well as OSS.

The most prominent example of pervasively deployed OSS is the "LAMP" Web server stack, which comprises the Linux operating system, Apache Web server, MySQL database management system, and the Perl, PHP and Python scripting/programming languages. The wide adoption of the LAMP stack by major enterprise customers has helped credential the commercial viability of community developed software, and accelerated the development and update of other open source programs. The increased credibility of such OSS has also inspired major vendors, such as IBM and (more recently) Sun, to base more of their strategic direction on the provision of services based in part on OSS, rather than on the sale of their historically proprietary products.

As already noted, however, there have been increasing patent challenges to both OSS and open standards. The ongoing litigation by SCO, which claims (among other things) that elements of its proprietary software have been included wrongly in Linux, cast a temporary shadow over that operating

system and OSS in general. Additionally, so called "trolls" have asserted their patents against implementations of a variety of standards after those standards became widely deployed.

The result has been a heightened sensitivity to intellectual property rights (IPR) on the part of both IPR owners as well as users of OSS and open standards. This has resulted in a rude shock for the open source community, which has increasing needs to utilize open standards in order to integrate OSS interoperably into mixed end-user environments. Because many of those open standards are created under traditional "RAND" terms (i.e., "reasonable and non-discriminatory terms"), this may be impossible, because RAND terms allow IPR owners to both charge royalties as well as forbid sublicensing of rights conveyed.

There is a second challenge to OSS, however, that has been less recognized. That challenge arises from the dynamism of OSS itself. At the community level, OSS is under constant, real-time evolution. And at the user level, OSS licensing terms permit the making of whatever changes are desired, whenever the user so desires. Open standards, in contrast, only work when certain elements of a software program are kept constant, or are only allowed to change in ways that would not jeopardize interoperability.

Of course, an open source community can make it a priority to maintain interoperability, and simply avoid making changes that would break compliance with a desirable open standard, and an end-user can make the same decision. But what if the community is not so motivated, or puts a higher value on innovation in some area than maintaining compliance with the standard? And similarly, what if the developer of the standard decides to permit inclusion of IPR in its next release of the standard, and the owner of that IPR insists on requiring a license that is incompatible with open source licensing requirements?

The tolerance of open source licenses for the creation of derivative works that vary from the original causes a second problem that can undercut the value of open source software itself. This can arise when multiple distributions of the same OSS are each changed in ways that require the developers of other software to tailor their own wares to the unique requirements of each distribution. The result is that the owner of one such OSS distribution can no longer easily migrate to another nominally equivalent distribution without re-licensing or patching its other software. In short, the end-user has become just as "locked in" by the vendor of its OSS distribution as it might have been if it had licensed a proprietary alternative.

This is already happening in the Linux environment, where the distributions of Linux continue to multiply, each optimized in one way or another for a particular use, or to increase its appeal to commercial users. If left unchecked, the result would be a replay of the fragmentation experienced by Unix in the past, and the loss once again of the benefits to end users of a marketplace based upon multiple versions of a common operating system, none of which requires the end-user to lock itself in to the software and services of a single vendor.

The way to avoid this result is once again through the development of standards. Using Linux as an example, the standards create a layer between the operating system OSS and the applications that run on top of it. If each distribution vendor agrees to support the standard, then an end-user may change distribution vendors at will, with little additional cost associated with making the change.

In principle, the creation of such standards sounds simple. But in practice, there are challenges, because the open source community that is creating the OSS must buy in to the value of the open standards. If it does not, then it can evolve the OSS in ways that break backwards compatibility. Similarly, the developer of the standards must commit to continuing to track the development of the OSS wherever it goes, and its members must agree to forego royalties and restrictive licensing terms with respect to any patent claims that might be infringed by the developer's standards.

In short, things are not so simple, since both the OSS community and the standards developer must be aware of the constraints and goals of the other in order for the combination of the OSS and the OSS standards to have optimal value at minimum compromise.

Moreover, the role of the standards developer and the software developer are reversed in time. Rather than developers creating software to implement a completed standard, the developer of the standard is

creating a standard to keep pace with the ongoing development of the OSS – an altogether different and more challenging proposition.

The only way to achieve this result is for the two processes to proceed in parallel, and with a degree of communication that would not be necessary in traditional standards development. A partnership of sorts is therefore required, in which each participant recognizes the value of the other, and agrees to work in cooperation to achieve results of mutual benefit.

Today, there is a single example of such a novel partnership, and happily that partnership is succeeding. Not surprisingly, it was created to first address the danger of Linux fragmentation, and its name is the Free Standards Group (FSG). Given the fact that the risk of forking and lock-in is common to all OSS, the FSG example is worthy of study, and of emulation in any situations that the FSG does not itself plan to address.

The following interview is intended to showcase how the FSG has creatively bridged the historical divide between OSS and open standards and is succeeding in guarding Linux from suffering the fate of Unix. *[Disclosure: the author is a director of the FSG, which is also a client of his law firm.]*

The interview: This interview was conducted in May of 2006 by email with Jim Zemlin, the Executive Director of the FSG, and Ian Murdock, FSG CTO, chair of the FSG Linux Standards Base workgroup, and creator of the Debian GNU/Linux free Linux distribution.

I. Overview Questions

1. How did the FSG come about, and what was its original goal?

The Free Standards Group was formed in 1998 to promote open source software through standards. Specifically it was formed to prevent the fragmentation of Linux. The participants in the Linux ecosystem realized early on that an open standard delivering application portability was absolutely necessary to prevent the fragmentation of Linux. Out of this concern, the community banded together to form the Free Standards Group (FSG), a standards body tasked with developing open, international standards that would deliver on the vision of binary portability within a competitive Linux distribution ecosystem. Basically, we are trying to do for Linux what Unix tried to do and failed, resulting in Solaris, AIX, HPUX, Novell, etc…

2. How has that goal changed over time, and how would you phrase the mission of FSG today?

The goal has not changed, nor has the mission of the FSG. As the market for Linux has expanded and its importance to the industry has risen, so has the importance of the work of the FSG.

3. Please describe the LSB – what it standardizes and why these particular elements are important.

The Linux Standard Base (LSB) is an application binary interface (ABI) for Linux and Linux-compatible platforms. The LSB draws on the source standards of the IEEE POSIX standards and The Open Group's Single UNIX Specification for many of its behavioral interface definitions. Some interfaces are not included in the LSB, since they are outside the remit of a binary runtime environment; typically these are development interfaces or user-level tools. The LSB also extends the source standards in other areas (such as graphics) and includes the necessary details such as the binary execution file formats to support a high volume binary application platform. It includes written binary interface specification, a set of test suites for both distributions and applications writing to the standard, and a sample implementation for testing purposes.

Since it is a binary specification, the LSB is divided into both general and processor-specific components.

The LSB Specifies:
Common Packaging and Install Guidelines
Common Shared Libraries and their Selection
Configuration Files
File Placement (FHS)
System Commands
ABIs for System Interfaces (both application and platform level)

We expand our platform standard over time as the nature of the use of that platform changes. This change happens as new run times above the OS are introduced or new performance enhancements to the underlying architecture are introduced. The FSG concentrates its efforts in the following areas:

1. Developing and improving existing standards
2. Developing and implementing testing and certification programs in support of its standards
3. Conducing outreach and education campaigns to encourage ISVs to target the Linux platform, providing technical support and resources
4. Enforcing the LSB brand with compliant distributions and applications

4. To whom is the success of the LSB important, and why?

The success of the LSB is important to the entire Linux ecosystem including end users, ISV's, distribution vendors, systems vendors, and open source developers.

- End Users: The LSB is important to end users to preserve choice by achieving interoperability at the Linux OS platform layer. End users have always wanted to "make is as easy to get out of a particular platform as it is to get into one if I am not getting the support or innovation I require." As with any standardized component of a technical solution an choice among compatible implementations reduces cost, allows for easier support, and provides for a long-term support ecosystem that can be counted on. Specific users of Linux include government agencies who mandate choice in their procurement policies, large financial services companies who use cutting edge applications on Linux, and small to mid market users who want to look to a "Linux Standard" mark to indicate compatibility rather than resorting to guess work around what runs on which version of Linux.

- ISV's: Making it easier for applications to target the Linux platform is very important for the entire Linux community; this includes Linux distribution companies, hardware vendors and software vendors. The success of the LSB is also important for end users who would like to see a healthy set of applications for the Linux platform without being locked in to a single Linux distribution. Without a widely supported binary standard for Linux, a single vendor, de facto standard will emerge, effectively removing choice and locking end users in.

  It is important to understand platform market forces in order to understand the importance of our standard. An operating system is only as strong as the applications that run on top of it. While Linux presents unique challenges to software developers (including multiple distribution targets), it also affords them a tremendous market opportunity as the Linux systems vendors grow a massive market based on the shared invention of the open source platform. The Linux Standard Base (LSB) was created to eliminate much of the heavy lifting required by software developers (ISVs) targeting multiple platforms today. In other words, the LSB enables ISVs to cost effectively target the Linux platform, reducing their porting, support and testing costs while achieving a global market for their applications.

- Distribution Vendors: Organizations such as Red Hat, Novell, Debian, Ubuntu, Red Flag Linux in China, Turbo Linux in Japan and others. Linux distributions vendors benefit from the LSB in the following manner: First it enables them to collectively grow the ISV ecosystem for their platform. Second, it allows them to fulfill a promise of "choice" to their customers – something not taken lightly in the value proposition of the Linux brand. Third, it allows them to share the support burden of developing and supporting a mission critical OS across a community while preserving interoperability and backward compatibility. The Linux distribution community has found multiple ways to compete around this standard without "commoditizing" their business. In fact, by profiting from support and services rather than locking customers into a single solution the Linux vendor community has made substantial inroads in the market against Microsoft and Unix.

- Systems vendors: Companies such as IBM, HP, Intel, AMD, Dell and others are looking to provide their customers with a choice of solutions at the highest price performance level they can offer. While these organizations vary in business models and often have units that cross the spectrum of technical solutions, they are increasingly becoming high level solution providers and trusted advisors to their clients. In this world, shared support and development for a key underlying component of their business offerings is a highly valuable proposition. The existence of a standard which enables them to partner with a choice of providers of Linux systems is essential to preserving this choice.

- The open source community: The open source community is deeply concerned with the concept of freedom of choice in computing. To them this freedom comes from access to the source code. However, the community is becoming increasingly savvy to the idea that source code alone will not offer the freedom they desire and that the combination of a standard that enables islands of code to interact together and in turn grows the network community of users of that technology will eventually increase the number of collaborators in their shared development model.

5. What is the current status of LSB adoption, implementation and certification?

LSB adoption can be characterized as "at the beginning of a great adoption curve." Development and compliance with the core standard is complete and through the efforts of the FSG, the Linux Standard Base has achieved great strides in the past 24 months, including the following key accomplishments:

1. LSB 3.0 launched in 2005 resolving key issues between major distribution vendors (including key C++ libraries), resulting in the support of all major distribution vendors
2. LSB 3.1 launched in 2006, including support for portable desktop applications for the first time. All major distributions (including Asianux, Debian, Novell/SUSE, Red Hat, and Ubuntu) have committed to certifying to LSB 3.1.
3. The LSB workgroup now has a steering committee containing key stakeholders of the community, including senior engineering resources at Red Hat, Novell, Ubuntu and major ISVs. This directly translates into alignment of the LSB with the major distributions' release schedules.
4. The Free Standards Group board of directors now reflects key stake holders of the Linux ecosystem with senior members from such companies as HP, IBM, Fujitsu, Intel and Novell
5. For the first time, major ISVs such as CA, Veritas, Oracle, MySQL, BakBone and others have either joined the FSG and given their public support of the standard
6. We have engaged a short list of "masthead" ISVs with the goal of getting their applications certified to the LSB, and RealNetworks has already agreed to certify RealPlayer, its widely used media player.
7. New memberships have increased by 70 percent over the last year, including the addition of over a dozen ISV members where there was previously no ISV participation at all
8. Funding has nearly tripled over the last 24 months, allowing the organization to hire a professional management team
9. The Chinese Government has become a certification authority for the LSB and is using the LSB as the base of its emerging national standard for Linux; efforts in India are underway
10. The Department of Defense is now mandating LSB compliance in their procurement contracts; other end users are also planning on doing this

11. The FSG has experienced dramatically increased visibility and awareness of the standard through new marketing efforts, including features in The Wall Street Journal, Business Week, USA Today, eWeek and the Associated Press
12. The organization is now staffed with an experienced management team, including Jim Zemlin as executive director and Ian Murdock, creator of Debian, as CTO and workgroup chair of the LSB

6. What other standards will be important to complete in addition to the LSB?

The FSG currently works with other development groups like freedesktop.org to incorporate their technical work into the official standard for Linux.

7. Is the FSG unique, or are there other open source standards groups in existence?

The FSG is the only standards body devoted to open source. The unique nature of open source standards (standardizing the downstream implementation of disparate upstream projects) lends itself to a unique standards body which can effectively work with all members of the open source community.

8. The LSB was recently adopted as an ISO standard. What will that mean for the LSB?

The LSB was approved as a Publicly Available Specification (PAS) by ISO/IEC (the International Standardization Organization and the International Electrotechnical Commission), and is published as International Standard 23360.

ISO approval shows the world that Linux is a serious, mainstream operating system, a serious companion to POSIX systems. It provides a benchmark between procurement and vendor, preserving healthy competition without allowing fragmentation of the market. Standards have been shown to contribute more to economic growth than patents and licenses combined, and the LSB will open the door to Linux as a requirement in large-scale (e.g. Government) procurements. The approval of the LSB also makes it easier for individuals, companies and governments to concentrate their efforts on one unified program.

II. Structure

1. What interest groups are represented in the FSG today, and why is their active participation important?

There are four main groups who are represented in the FSG today.

- Distribution vendors. The distribution vendors are the enablers of the standard. Without their participation, the standard cannot achieve any success. And without their participation in the creation of the LSB, their support for it would be unlikely. As of April 2006, all major distribution vendors have pledged to certify on LSB 3.1, and senior representatives from each of them are part of the steering committee of the LSB workgroup. They will be attending the LSB Summit in June 2006 in Boston and helping to shape the roadmap of the LSB.

- ISVs. They will implement the specification in their development efforts and use the support provided by the FSG, eventually certifying to the standard. Not surprisingly, the Free Standards Group has also made good process with obtaining early support from many key ISVs: Oracle, Bakbone, VERITAS, IBM Software, Novell, Levanta, RealNetworks, MySQL, Hyperic and many others have recently engaged with the LSB in various forms (participation in the steering committee and LSB summit, certification, roadmap input, etc.)

  The Free Standards Group recognizes that different issues face ISVs of different sizes and is structuring its messaging and ISV outreach efforts accordingly. While larger ISVs have the clout to mandate which Linux platforms their customers must use, the situation is often the reverse for the smaller ISVs, with the larger customers mandating which Linux platforms the ISV must support (i.e., "we've standardized on Fedora Core 2, so you'll need to support that if you want our business"). Of course, those platforms will invariably differ between customers. Ironically, then, the ISVs that are least in the position to support a multitude of Linux distributions are the very ISVs that have little choice but to do so. For smaller ISVs, then, the LSB provides substantial

benefit, because it allows them to support a wide variety of distributions with a single build; and while the LSB may not provide 100 percent assurance of portability, less than 100 percent is often sufficient for the small ISV (not to mention better than they would be able to do on their own for any given distribution, given their often limited resources).

For larger ISVs, the value proposition of the LSB is substantially different, as the larger ISVs will never see validating to anything less than a full implementation as viable. However, given the diversity of the Linux ecosystem and the disproportionately large potential upside available to the larger ISVs, these larger players are often challenged to find a way to support regional- or market- specific Linux variants. They still would like to broaden the market for their Linux products while minimizing the cost (and risk) associated with supporting additional platforms, which is necessarily much greater than for the small ISVs. For example, while Oracle formally supports only Red Hat, Novell/SUSE, and Asianux, it makes clear that its products work on Debian too (because many regional variants are Debian-based), though use on Debian is officially unsupported. Here, the LSB provides the larger ISVs with a clearer way of offering some assurance of functionality with the "long tail" of distributions but also makes clear that this assurance is less than full support.

- End Users. End users are the tailing adopters of the standard, yet they can assert significant influence on both distribution vendors and ISVs. End users need to see the LSB as a form of risk management in their Linux strategy. This is the demand from end users to fulfill the promise of choice. There is proof that acceptance has begun with early adopters. A handful of large Fortune 500 companies have stipulated LSB compliance in their procurement policy, license and support contracts, including the Department of Defense. More need to follow.

- Open Source Community. The open source community represents an amalgamation of software projects which are integrated into a single computing solution. A fundamental flaw in the open source development model is the lack of coordination across projects. There is good internal coordination among but not across projects. Because Linux distribution is made up of a broad set of disparate projects, different projects need to be coordinated effectively to achieve interoperability and backward compatibility across distributions. It's important that maintainers of these projects remain involved in the LSB process, track downstream implementation of the use of their software and give feedback to the roadmap of the downstream standard. It is important that the maintainers of those projects are aware of existing computing standards such as the LSB so they can work in a cooperative fashion to accelerate the adoption of their technology. The LSB and activities of the FSG is the best type of organization to coordinate these activities. Currently, key members of the community are involved in the LSB workgroup and will be attending the LSB summit.

2. How is the Board structured, and why?

Currently the Free Standards Group has three classes of membership. Financial commitments vary, according to the type and level of membership:

1. Individual Membership: Open to any person who is interested in supporting free and open source software.
2. Nonprofit: Open to any registered nonprofit organization that is supporting free software and open source software. This class of membership is also open to educational institutions.
3. Corporate: Open to any commercial entity engaged in the production, manufacturing, support, development, or sale of products supporting free and open source software.

In many ways, the Free Standards Group is structured much like consortia that traditionally create open standards. It has one major distinction, however, since it has to balance the needs of the open source community instead of purely being structured to advance corporate interests. Board members are elected by members and represent each of the three classes of membership. Directors are elected for a two-year term.

3. In what ways is the structure of FSG different from a typical open source project?

The task put before the Free Standards Group is a complicated one: it must bring “cathedral” structure to the open source “bazaar.” In order to be relevant to the distribution vendors and to ISVs, deadlines must be met and the specification must be current, timely and up-to-date. The FSG is closer in many ways to a commercial company than to an open source project. An open source project can iterate its software early and often, constantly updating and adding features as implemented. A standards body, however, must have predictable, publishable release cycles that are in synch with the distribution and software vendors who implement the standards. In order to meet these demands it also cannot be staffed 100 percent with volunteers. Key technical leadership and project management/consensus building roles must be full time FSG staff.

4. How many paid staff are there, and what are their roles?

There is an executive director (Jim Zemlin) responsible for the overall direction of the FSG. A CTO (Ian Murdock) who is responsible for the technical direction of the FSG (and who is also the chair of the LSB workgroup). The organization has a director of marketing/communication (Amanda McPherson) responsible for marketing, communication and outreach activities. The organization currently employs one engineer and a staff in India and plans on adding more for core LSB work.

5. Approximately how many individuals are involved at the technical level, and what do they do?

There are currently over a dozen individuals devoted full time to the technical work of the FSG, with many more involved on a part time basis. These people are employed by such member companies as Intel, IBM, HP, etc and work on the specification, test suites and certification programs.

III. FSG – Linux Interface

1. Which open source projects does FSG actively engage with?

Primarily the Linux distributions but also many of the constituent projects, particularly if those projects provide a platform that developers can target that could benefit from better integration with the broader Linux platform. Good examples here include the GNOME and KDE desktop environments. Each of these desktop environments is a platform in its own right, but a desktop isn't much use unless it is well integrated with the operating system underneath. Furthermore, ISVs targeting the Linux desktop ideally want to provide a single application that integrates well regardless of which environment happens to be in use.

2. How does FSG work with the Linux development team and the Linux process?

Actually, the LSB doesn't specify the kernel--it only specifies the user level runtime, such as the core system libraries and compiler toolchain. Ironically, then, the _Linux_ Standard Base isn't Linux specific at all--it would be entirely possible (and probably not altogether hard) for Solaris to be made LSB compliant. The LSB is entirely concerned with the application environment, and the kernel is usually pretty well hidden at the application level.

3. Does the Linux community participate in FSG as well?

Yes, though most participation comes from engineers that work for the various companies that have an interest in Linux (Intel, IBM, Novell, HP, Ubuntu, etc.). However, there's nothing particularly unusual about that. Most open source development these days is done by commercial interests, not by college students working out of their dorm rooms, which seems to be the common perception. (Of course, a lot of it starts there, but the best developers eventually figure out how to get paid to do it.) Whether you're interacting with paid engineers or unpaid volunteers, though, a key to success in the open source community is getting the right people to buy in to what you're doing and, ideally, getting them to participate. In general, the FSG mission resonates well with the open source community, and we have little difficulty getting that buy in and participation.

IV. FSG – Linux Dynamics

1. I've heard you describe the relationship of the open source and open standards processes in "upstream" and "downstream" terms. Given that open source development is "real time" and ongoing-release, while standards have traditionally operated on a fixed basis, with nothing changing for a period of time, how do you make this work?

One way to understand this is look at the attributes of a successful open source project. Success is relative to the number of developers and users of a particular set of code. Apache is a good example. As the community iterates code with similar functionality, for example a web server or a C compiler, the participants end up aligning themselves around one or in some cases two projects. Smaller projects tend to die. The ones that succeed then join the many other packages that are integrated into a platform such as Linux.

The trick in standardizing then is to decide which snapshot in time – which interfaces from those packages at that point across all these packages - will guarantee interoperability. By coordinating with these disparate upstream projects which versions of their code are likely to be broadly adopted downstream with the distro vendors, we provide a framework for those working both upstream and downstream. In the case of the Linux distros, we help them cooperate in order to bring meaning to the term "Linux" in terms of the type of interoperability that is commonly expected on an operating system platform such as Windows or Mac OS.

This effort requires ongoing awareness of the spec development process itself both upstream and downstream, and a rapid feedback framework for all parties. It also requires a coordinated parceling of the testing efforts to the appropriate sub-projects. In other words, we are applying the bazaar method of open source coding to the development of standards. That is how the community plays and we are a part of that community.

2. At the process level, what other aspects of open source development are most problematic for standard setting, and vice versa?

Before answering that question, there's one very important thing to understand about the FSG, and that's that we don't define standards in the same way that a traditional standards body defines standards. And that's just the nature of the beast: The open source community is vast, complex, amorphous, and continually in motion. It's also an integral part of what we do. So, the FSG by nature isn't just a well-defined consortium of technology vendors that can define things unilaterally. It's a well-defined consortium of vendors, certainly, but it's also more than that, in that the vast, complex, amorphous, continually moving open source community needs to be represented at the table. In a lot of ways, what we're doing at the FSG, namely bringing together open standards and open source, is unprecedented.

Clearly, our interactions with the open source community affect the processes we use to build the LSB and our other standards. We can't just say "this is the way things are" the way we'd be able to do if our constituency was smaller and more self-contained. Instead, the way we define standards is far more about consensus building and observation--we watch what's happening in the open source community and industry and track what's emerging as a "best practice" through natural market forces and competition.

One of the challenges of the LSB project, then, is understanding what technologies have become or are becoming best practice, so that we can begin the process of incorporating those technologies. Another challenge is dealing with a moving target--after all, although the process of defining the standard is different, at the end of the day, the standard has to be every bit as precise as, say, a plumbing specification, or it won't guarantee interoperability. Fortunately, we already have a model to follow here, namely the Linux distributions, which perform the analogous task at the technology level by assembling the various open source components into a cohesive whole.

So, our task essentially boils down to tracking the technologies that ship in the majority of Linux distributions, and in building a layer of abstraction, a metaplatform of sorts, above the multiplicity of

distributions so that application developers can target a single, generic notion of Linux rather than each distribution individually.

We also work to increase participation in the ongoing development of the standard and to facilitate collaboration among the key stakeholders to more rapidly reach consensus around the best practices. The goal here is to capture in the LSB roadmap not just what exists in the current generation of the major distributions, but what's coming in the next as well. After all, ISVs developing Linux applications today will often see the next generation as a primary target.

3. What compromises (technically and process-wise) have the Linux and FSG communities had to made in order for the LSB to be practical while not impeding the work of either side?

The biggest challenge in what we do is probably no different than in any other standardization effort: Balancing the need for standards with the need for vendors to differentiate from each other. However, in the open source world, this tension is probably more pronounced due to the speed at which development occurs. I'd say the biggest compromise the open source community makes is understanding the importance of standards, backward compatibility, and all the sorts of things that tend not to be "fun" but which are vital to commercial acceptance--and being committed to doing what needs to be done. On the FSG side, the biggest compromise is being fairly hands off and leaving it to the marketplace to determine which of many alternatives is the best practice. The real key is making sure interoperability problems don't crop up in the process, and the key to making sure that doesn't happen is ensuring all the parties are in a constant dialogue to make sure the right balance is struck. We see that as one of the roles of the FSG--providing a neutral forum for these kinds of conversations between the key stakeholders.

V. Looking to the Future

1. Where else are organizations modeled on the FSG needed?

I wouldn't frame it as where else is an FSG needed but rather where should the FSG go from here? At the end of the day, the LSB is a development platform standard. Some developers target the operating system in C or C++; others target middleware platforms like Java or LAMP; others are moving further up the stack to the web, where applications span site and even organizational boundaries (think of the various "mashups" that are happening around the so-called "Web 2.0" applications like Google Maps). Today, we cover the C/C++ level pretty well, but we need to move up the stack to cover the other development environments as well. The ultimate goal is to provide an open standard developers can target at any layer of the stack that's independent of any single vendor.

So, the short answer is that we aspire to provide a complete open standards based platform ("metaplatform" is actually a more accurate way to describe it), and Linux is obviously just one part of such a platform. We need to move up the stack along with the developers to incorporate the higher level platforms like Java and LAMP. We need to extend the coverage of the operating system platform too, as we've done in LSB 3.1 with the addition of desktop functionality and are doing around printing, multimedia, accessibility, internationalization, and other areas in LSB 3.2. Even at the operating system level, there's nothing inherently Linux specific about the LSB, so there's nothing preventing us from encompassing other open platform operating systems, such as the BSDs or Solaris. In the end, it's about all open platforms vs. closed platforms, where the closed platform du jour is Windows.

So, the real question is, how can the open metaplatform better compete against Windows? For one, Windows has .NET. Linux (and the other open platform operating systems) have Java, but it's not as well integrated, and it's not as well integrated because of the Java licensing. Sun has indicated they're going to open source Java as soon as they address the compatibility concerns. We have a lot of experience in that area, so perhaps we can help. In the end, it all comes down to a strong brand and tying compatibility testing to the use of that brand, which is the approach we take with the LSB. There's no reason a similar approach couldn't work for Java, and the benefit of a integrated Java with the open metaplatform would be enormous.

Obviously, doing all of that is an enormous amount of work, undoubtedly an impossible task for any single organization to accomplish on its own. Then again, so is building a complete operating system, and a lot of little companies (the Linux distribution vendors) managed to do it by taking preexisting pieces and

fitting them together into integrated products. And, as it turned out, the whole was a lot more valuable than the sum of its parts.

We take the same approach on a few levels. First of all, the LSB is an open process, so the best way to get something into the standard (assuming it's a best practice, ie., shipping in the major Linux distributions) is to step up and do the work (i.e., write the conformance tests, etc.). In other words, we leverage the community the same way an open source software project would. Second, there are a lot of open standards efforts tackling pieces of the overall problem, and we seek to incorporate their work. In that sense, we're essentially an integrator of standards, a hub of sorts, much as the Linux distributors are essentially integrators of technology. We don't have to solve the total problem ourselves, just provide an open framework in which the relevant pieces can be fitted together.

2. In the long term, should the standardization process and the open source process merge? In other words, is there a benefit to there being an independent FSG, or in the future would it be better if the open source community incorporated this role into its own work?

Currently, there is no better way to balance the needs of a competitive distribution community with application interoperability. An independent standards provider bridges the gap between the open source community and the distributions implementing their software by allowing best practices of the latter to be standardized, thus making it easier for ISVs and end users to actually use the platform. The open source community does not want to concern itself with this standardization concern, nor should they. An independent consortium can drive consensus while being politically sensitive to the needs of its constituents.

3. What is the single thing that open source advocates most need to "get" about standards, and need to work harder to accommodate? Same question in reverse?

It would be helpful if those in some of the upstream projects participated more closely with our standards efforts. They are already doing this but there is always room for more participation. Closely tracking of the projects into the standard (or just through a database) will provide a great deal of service to ISVs and the distribution vendors. We plan on offering this service.

In the other direction, standards bodies need to recognize that open source development is fundamentally different than traditional software development. When working with the open source community, participation and buy-in are critical—you can't just declare something to be so and expect the open source community to just follow suit—as is the ability to move quickly. For the FSG's part, we understand all of this very well—after all, we grew out of the open source community—but it's an observation other standards efforts would do well to keep in mind as open source and open standards increasingly intersect.

VI. Open Ended

What haven't I asked that I should have to complete this picture?

Andy, I think it is worth noting that Linux is different from Unix or Microsoft Windows where the brand has a rigorous testing and certification process. the Linux brand is owned by Linus Torvalds who does not choose to enforce that brand through any kind of technical compliance mechanism. This is creating an oncoming crisis for Linux as differing groups seek to define the brand.

The computing industry has made an enormous bet on Linux. By many accounts the total investment in research, support, development and marketing of the Linux brand exceeds many billions of dollars. This investment is spread predominately over a few “computing giants” who stand to profit from the culmination of products and services sold in support of the Linux operating system. By all accounts, the rise of Linux has been precipitous, creating wealth for the ecosystem while spreading the risks among those who shoulder the burden of development and marketing.

While there has been an enormous investment in the Linux brand, the un-enforceability of the Linux trademark introduces a great amount of risk for those who have made financial bets on the positive impact of the name. As Linux evolves from the world of the data center to the world of the desktop and

mid-market, customer expectations will also evolve. Data center customers expect software to require customization to work in their environment. Mid-market, small businesses and end users, however, have different expectations. They want their Linux application to run on their Linux operating system. If it doesn't work "as promised" by the industry giants who have created the Linux brand, the customers will experience Linux as broken or poor performing. A negative halo effect will ensue for those vendors who have supported or sold products on that system. Who will benefit the most in this scenario? The makers of the Solaris and Windows operating systems.

Many would say there is a solution. A Linux distribution (or sub-brand) can guarantee applications run on its version and thus make the guarantees to the customers (and ISVs). As long as the systems vendors and ISVs work with the right sub-brand (distribution) everything will work. While in many ways this has worked so far in the history of Linux, there are risks associated for the Linux ecosystem.

A successful sub-brand subsumes the Linux brand for itself. As long as others in the ecosystem remain committed to the sub-brand, all is well. But as the power of the sub-brand rises, its ability to hold its business partners hostage with unreasonable licensing, business or financial demands does as well. There is a long history in the computing industry of sub-brands (especially software companies) accomplishing this feat.

There is also risk that the sub-brand will evolve up the software stack to compete directly with its business partners. Or, as has been discussed widely in recent media coverage, a large competitor could acquire the sub-brand and change the Linux ecosystem over night. For an industry with a multi-billion-dollar investment in Linux, this risk is unacceptable. At the most this risk is something that should be stopped, but at a minimum this risk is something that should be managed -- especially when the cost associated with that risk management is relatively low. A well supported open standard – the Linux Standard Base – and attendant developer out reach and education is the best way to manage that risk.

Comments? updegrove@consortiuminfo.org