



---

## **Open Standards - Open Source**

**The Business, Legal, and Technical Challenges Ahead**

June 24-25, 2003, Minneapolis, MN, USA

---

# **Proceedings**

THE *Open* GROUP

Copyright © 2003, The Open Group

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior permission of the copyright owners.

“Open Standards – Open Source: The Business, Legal and Technical Challenges Ahead”

Published by The Open Group, July 2003.

Any comments relating to the material contained in this document may be submitted to:

The Open Group  
44 Montgomery St. #960  
San Francisco, CA 94104

Or by Electronic Mail to:  
[os-business@opengroup.org](mailto:os-business@opengroup.org)

## Summary

The meeting was held June 24-25, 2003 at the University of St. Thomas (which also co-sponsored the event) in Minneapolis, Minnesota. The one-and-a-half day conference brought together senior executives from a wide range of organizations to discuss and evaluate the current state of Open Source. Representatives from Government, Business, and Academia were all on hand to offer unique perspectives.

The meeting comprised four panels: Business, Technical, Legal, and Social and Ethical, each of which featured an introduction of the issues and follow-up with an interactive discussion between the speakers and the audience. The aim was to capture and publish the issues discussed in order to raise the industry awareness of the benefits of Open Source.

The four panel discussions examine the issues surrounding Open Source from various perspectives:

### Business

- Making the business case for Open Source
- The business drivers for Open Source
- What is the future of Open Source?

### Technical

- Is it robust, scalable, portable, interoperable?
- Is it feasible for SME's (small, medium enterprises) to use Open Source?
- What is coming down the road?

### Legal

- Understand licensing and distribution
- Managing intellectual property rights
- Are there any special considerations for intellectual property rights?

### Ethical & Social Implications

- The ethics and code of conduct for Open Source developers
- The social ramifications of an Open Source work force
- What impact will Open Source software have on the IT divide?

The meeting dug into the significant issues in all four areas, with collaboration models dominating the discussion, even though contentious issues hit every panel. The focus of each panel was the quantifiable benefits and the steps to risk mitigation by relying on Open Source, as well as perspectives on the enormous social and ethical implications of the Open Source movement.

***Keynote: Open Source as a Tool of Boundaryless Information Flow***

*Allen Brown, President and CEO, The Open Group*

We designed this conference specifically for a group of this size and this make-up—an amazing mix of academia, large customers, vendors, small organizations, and developers. What we want to do today is not have a conference where people are just speaking at you, but to engage you on many levels.

To start, I'll do a little bit of context setting, to talk about what The Open Group is doing and why we care about Open Source. Then I'll explain how we're going to move through the day.

We talked to a lot of CIOs, so I hope that this resonates with the customers in the room. What CIOs tell us is that they are under a significant amount of pressure. And that pressure is coming from the changes within their organization.

The pressure is to be able to deliver information when and where it's needed in a timely and reliable manner. Why is the pressure on them? The problem is that organizations are changing and they need to be able to reliably deliver integrated information when and where it's needed. That has required changes in when, where and how information goes from one person to another. So their key issue is integrated information and access to it—they've got to be able to provide that to people.

We took this finding to our Customer Council. The Open Group is a consortium—a partnership between vendors and buyers—and the Customer Council, naturally, reflects the thinking of members who are buyers. These members worked on something called the “Interoperable Enterprise”, which is a Business Scenario [The methodology we use for describing a problem in a business context] available on the web free of charge. That Business Scenario describes the pain that large customers have with delivering integrated information. Let me describe the issues it covers.

Enterprises were originally established in an end-to-end process. They buy raw materials, make them into something, and they sell them. For many years, we organized people into departments in a particular field, and within them, started to get situations where people would become experts.

There were two great benefits to doing it this way. Departmental experts could do things faster because they did things more repetitively, and they could improve quality because they built best practices for their particular area.

But what resulted is what we all know today as stovepipes or silos. So if you look around, you find that within an organization we developed a lot of stovepiped, or siloed, departments. We'd even go so far as to call some parts of our organization “divisions.” Over the last ten years or so, most of us have been trying to break down those silos and stovepipes. Whether we tried business process re-engineering or other approaches, the person who really pushed this home was Jack Welch at GE.

Do you know the story of what Jack Welch did at GE? It's called the Boundaryless Organization. What Jack was concerned about was the horizontal, vertical, and geographic barriers within organizations that prevented the organization from working. In today's environment, where we have to react much more quickly to any situation, we don't want things going in a departmental hands-off process. We want to get people together to attack problems quickly and efficiently. And that's what we've been doing with cross-functional teams that come together specifically to solve a particular problem or set of problems.

When we started doing this, the big problem was the culture between the people in the different organizational stovepipes. They didn't talk to each other. Hopefully, we've overcome that now in most of our organizations.

[By the way, when I give this presentation in government, I hear, "If you think you've got stovepipes, come and work in the government. We'll show you stovepipes. We'll show you silos."]

Having overcome the barrier between people, we've found that within those stovepipes or silos the IT systems were specifically built for the silos. They were all conceived individually and weren't intended to work with others. But we're not just going to throw away systems that have been built up over perhaps 20 or even 40 years. The challenge gets even more complex when we bring business partners into the picture.

To get a sense of the scope of this hurdle, we asked one of our customers two questions: "How many business partners do you have?" and "How many applications do you have?" They replied "Over a thousand," and "We don't know."

Imagine trying to find information, putting it together, and then delivering it where it's needed to cross-functional teams that are continuously forming and re-forming. You can't just put in a new application for each team, because by the time you study their specific need and respond to it, the team has gone away and others have replaced it.

As Jack Welch called the vision for the whole enterprise the Boundaryless Organization, we focused on a vision of Boundaryless Information Flow—getting information to flow in a boundaryless way within the organization. And the added complication for these organizations is that they're global and have to get interoperability working at a global level.

If we take a look at The Open Group's vision statement, we see something that says, "Boundaryless Information Flow achieved through global interoperability"—but that's not enough. Jack Welch himself said that boundarylessness does not mean having no boundaries; that would be silly. But what we must have are effective boundaries that are appropriate to the business need. What we don't need are boundaries that disable business. We need boundaries to be permeable so they *enable* business.

If you talk to members of our Security Forum today they are tackling security in a way that enables business, not prevents it. It's the same story with our other Forums and working groups, like Quality of Service. What we are trying to achieve is to help industry move towards a position where we can have Boundaryless Information Flow through global interoperability in a secure, reliable and timely manner.

By the way, the book on the Boundaryless Organization was written by the consultants that worked with Jack Welch on the GE Work-Out program. The lead author is Ron Ashkenas, who is going to be joining us at our Boston conference for a lighthearted presentation at a cocktail reception.

Just as Jack Welch could not go out and buy a Boundaryless Organization, neither can The Open Group make a Boundaryless Information Flow package that you'll go out and buy. It's a continuous process. You have to work through all the variables, just as our different Forums are doing now.

What we do is bring customers and suppliers together to understand what the issues are, to understand what the pain is and what the requirements are. We aim to understand where we need standards, where we need tailored or other solutions and to enable the industry to deliver the standards and solutions. It's important to make sure that we know what we're getting.

One of the big issues with standards is that customers don't necessarily see the benefit. They hear from a vendor, "My product complies with standards. Trust me—I'm a salesman." And they know that doesn't tell the whole story. The Open Group does certification of conformance to standards. It's only by certification of conformance that customers know that the products will work according to a standard.

That takes us to some of the things that The Open Group does. Business scenarios (I've already mentioned the Interoperable Enterprise), Architecture Framework, Challenges, Specifications, Certification, Best Practices, Open Source. The reason we have to deal with all of these things is that in order to help industry get to Boundaryless Information Flow, we can't just have one tool in the toolbox. We can't just go round with a hammer thinking that every problem is a nail. We've got to have the appropriate tool for the job.

There are a couple of Business Scenarios that our members have done in addition to the "Interoperable Enterprise", we have Identity Management, and the "Executive on the Move" on the web site. They are freely available.

With "Challenges", members come together and say, "We've got a specific problem with existing products that we need to get fixed now. Are there ways of configuring products to do that?" The latest example was concerned with the need for security of messages using e-mail clients within and between organizations. This resulted in the Secure Messaging Toolkit. The Mobile Directory Challenge is next.

Open Source is another tool. And Open Source is not just about the platform. An example of how we use this tool is OpenPegasus.

Over the years, we have had a real problem with management systems. Management systems did not interoperate and we had working group after working group of vendors sit down together to try to figure out how their products could interoperate. They never arrived at a solution. Every couple of years they'd come up with a plan and say, "We've come up with a great idea on how this is going to interoperate." And after the initial enthusiasm died down, they'd realize that it was one particular vendor's solution and it would cost a fortune to re-engineer. What happened with OpenPegasus is that the vendors came together and agreed upon a Common Information Model (CIM) that needed an implementation for which we had one source. This gives us a translation layer—a layer that will enable us to communicate from different management systems. So what we've achieved with Open Source in this case is to overcome an intransigent problem that specifications alone couldn't achieve. We couldn't reach consensus any other way.

In the future, as this is going back into product, we want to make sure that it goes back into product in a consistent way so we do need a consistent standard or specification. And then at some stage customers are going to want to know that the products they're buying are conformant with that specification. So then we would want to see a certification program.

Turning to specifications and standards, let me offer a couple of examples of implementations that The Open Group has been developing. You may have seen a press release from Seibel talking about ARM, Application Response Management, going into their products right now. Our Enterprise Management Forum worked on ARM. We also have a key role in the Single UNIX® Specification. Some of the questions in this area include: Is a specification a standard? Who sets standards? Are standards only done by formal standards organizations or do consortia do them? Do consortia do recommendations, do they do certification, what do they do? These are related issues we deal with constantly. Our position is that mostly what we do are specifications and the formal standards bodies do standards. So if you look at the Single UNIX Specification, although that is regarded as *the* standard in the industry, in actual fact, it is a profile and a whole list of specifications.

Does Open Source equal open standards? I've been to EC meetings where the representatives of the different countries all sit there with their little flags and the experts sit in the middle. Afterwards, at a recent meeting, one delegation came up to me and said, "We are very confused. You seem to suggest that Open Source isn't a standard." But it *isn't* a standard. If it conforms to a standard, it's standard. But Open Source as a methodology doesn't develop a standard product any more than proprietary development.

Open Source can only be standard as a methodology. Linux, for example, is not a standard until people agree on a standard. In this case, the Free Standards Group

developed the Linux Standard Base to try to make sure that a standard comes together for Linux.

There's a paper that you might want to see from EURIM, which is the UK government advisory on their work on Open Source. One of the first issues they address is, "Is Open Source a standard?" And, "Is Open Source a standard model?" That answer to the second of these is yes. It is a methodology and a process for work.

The next tool we can look at is certification and testing. This is a big part of The Open Group life. Many people know, or are getting to know recently because of certain events, that we own the UNIX® trademark and that we certify server products and thereby enable them to use the UNIX name. People are not specifically concerned about the servers, they are concerned that if they're using UNIX, especially in a procurement situation, that the products conform to the Single UNIX Specification and have been properly certified as such.

We also stand behind the Linux Standard Base certification. One of the things that happened there was that we donated about 95 percent of the test suites to that activity so that Linux Standard Base certification could happen. Additionally, we are involved in CORBA; Wireless Application Protocol; LDAP, which is an IETF specification; Schools Interoperability Forum, which is a completely different vertical activity; and Digital Video Broadcasting Multimedia Home Platform, another area where we're doing testing. One more tool in the tool chest.

The final one is best practice, where our members come together and develop things like the Manager's Guide to Information Security. It's truly a manager's guide, not something on a technical level. It's another tool. Our members are now working on a Manager's Guide to Open Source.

Open Source and standards are among the tools that we will use in order to get to Boundaryless Information Flow.

This is a unique event; I don't think anyone has assembled a group of this quality before and considered not just the technical issues, but the business, the legal, and the intriguingly, the social and ethical issues.

In September, we will be putting on another conference on the subject of Open Source in London. It is likely to be very different, however. In Europe about 80 percent of all businesses are small and medium enterprises. And right now, they're not that interested in the subject of Open Source because they don't understand how they can use it. They haven't got people that can download Open Source software and integrate it into their operations, and we don't know where to turn. That is one way of saying we see an immediate need for whatever the people in this room learn and share during this event.



### ***Rules of the Conference***

- Panelists limited to 10 minutes up front
- No overt bashing or flaming
- Interactive discussion of at least equal length after the presentations
- Have fun

### ***The Business Panel***

*Moderator: Graham Bird, Vice President of Marketing, The Open Group*

*Panelists:*

*Andrew Aitkin, Olliance Group*

*Loren Sinning, Cargill, Inc.*

*Carolyn Kahn, The MITRE Corporation*

*Stormy Peters, Hewlett-Packard*

Graham Bird, Vice President of Marketing, announced The Open Group's Open Source Project, chaired by Walter Stahlecker of the Hewlett-Packard Industry Standards Program Office. Founding members of the Project were the impetus behind this Open Source event in Minneapolis. It is important to note that The Open Group already has about 200-300 organizational members of varying sizes and 2,000-3,000 regular participants in conferences, working groups, and related activities. A conference such as this is designed both to energize and to educate that membership about Open Source as well as contribute to the activities of the Open Source community and engage new companies in The Open Group's programs.

Graham posed questions that set the stage for the Business panelists and the ten-minute presentations of their points-of-view. How do you persuade business colleagues to give Open Source a serious look? How do you generate confidence in them to try Open Source solutions? How do you confidently bet your job on Open Source?

Noting that there were business considerations that crossed over into technical and legal issues, he plowed forward by taking an advocacy position about Open Source: "I think Open Source is a good thing. I think it's an important thing. I think it's an inflection point of this industry." He then challenged the audience to respond to the presentations with a list of top-level issues that would, if addressed, justify his confidence and "make Open Source accessible in the enterprise." The Open Group is one of the places where producers and customers can achieve clarity about what they want to achieve and where changes can be made.

"We are not doing this because Open Source is good, because we like it, because it's fun—although that's true," Graham said. "We're doing it because we run a business and we'd like to eat tomorrow. It's a business-focused objective."

With a nod to the importance of the platform, Graham admitted that his real interest was on the mission-critical applications. He referenced the work of the EURIM group [see Appendix A], noting that one of the participants at a recent meeting had said that the (European Community) governments should drive all its work through the Open Source

model. Is a suggestion like that way out of line, or a harbinger of government—and corporate—IT operations?

“What do you need in Open Source products and services to run your business? Tell us. Ask us. This is my number one: How do we deliver that confidence into an organization, into a management team? What will enable us to commit mission-critical applications to Open Source?” One of the key issues is cost reduction, but certainly not the only one.

A final question to launch the panel was “Should businesses care about standards? Should they feel compelled to influence the requirements expressed in standards?”

**Andrew Aitkin, Managing Partner and Founder of the Olliance Group**, presented first. Olliance Group helps clients achieve a competitive advantage and economic gain by leveraging Open Source technologies. An independent, Open Source strategy-consulting firm, Olliance works with customers such as Intel and Nokia to help them develop Linux and Open Source products strategies, as well as with end-users, to whom they offer education, business strategy, and technology planning services.

Andrew began with his view of the commercial ecosystem that surrounds Linux and Open Source. In surveying the audience, he found that about one-third of the attendees represented technology vendors, the people who make the products and solutions around Open Source. The balance represented users, including academics and self-designated small users, with an even split between active contributors to the community of Open Source and people who are new to Open Source.

The growing popularity of Open Source is having a profound effect on many technology vendors. Historically, Intel has driven the bulk of its revenue through its relationship with Microsoft. That’s beginning to change. Intel understands that, over time, Linux will be a core driver for them, so they have to walk a fine line between protecting their current relationship, yet beginning to work with a number of other partners on developing Linux solutions on IA architecture.

One of the challenges they are going to begin to face is that, up until recently, they thought they had a wonderful strategy where they could continue to own the desktop through their relationship with Microsoft, and yet make significant strides in the data center working with firms such as Unisys, HP and others. The issue that they’re going to face is that the adoption of Linux on the desktop is coming much faster than they—or many others—realized.

The issue with application vendors—and this is the core to the adoption of Linux—is that the Tier One vendors such as Computer Associates and Oracle understand Linux and are beginning to understand Open Source. However, the Tier Two and Tier Three partners of those companies, such as vendors serving vertical markets like financial services or insurance, do not understand this space yet. They may have read the reports and see that an opportunity is emerging, but they don’t grasp how it will affect them. They don’t have a sense of what it will cost them to enable a product to run on Linux, to invest in sales

and marketing to reflect a new strategy, how much they will have to invest in maintenance support and training, and so on. When they understand those issues better, adoption of Linux and other Open Source applications will come faster.

The other component of the ecosystem that has a challenge is the resellers and integrators, who are in a spot similar to the Tier Two and Tier Three vendors. They have been happily providing services around proprietary technologies. Today, some of them have silos, or specific instances where they are providing support for a Linux or Open Source solution to a particular client in a particular segment, but they do not generally make those services available.

So the issue for end-users is, “Where do you go for support?” IBM, HP and other major vendors provide it, but it is not yet universally available.

The following names refer to viable, robust Open Source solutions that are available today:

- Apache
- JBoss
- Sendmail
- OpenOffice
- Linux
- Zope
- Samba
- MySQL

Behind each one is a company that provides skilled support services. A customer could have confidence that deploying one of these solutions will involve maintenance, training, customization, and so on.

Andrew concluded his presentation with an overview of the process of developing and deploying a solution, with an emphasis on the four, most important ones if the solution is Open Source:

- Identify gaps, planned upgrades, or high cost/low return solutions in current IT architecture
  - *Identify and qualify Open Source software that meets company IT objectives*
- Estimate return on investment (ROI) and total cost of ownership (TCO) of migrating specific platforms and functions to Open Source.
  - *Obtain executive management approval and sponsorship for planning Open Source initiatives.*
  - *Identify key executives and technologists to lead Open Source efforts*
- Plan a series of projects to integrate Open Source software and methods into the IT infrastructure.
  - *Initiate research and communication with the Open Source community.*
  - *Develop an Open Source Policy including guidelines for using Open Source software and engaging with the community*
- Conduct Open Source migrations

**Loren Sinning, Senior IT Advisor for Cargill, Inc.** brought the perspective of a large user. Cargill is an international marketer, processor and distributor of agricultural, food, financial and industrial products and services, with 97,000 employees in 59 countries. It has more than \$51 billion in gross sales and, in terms of revenues, ranks 20 in the Fortune 500. With the company spread all over the globe, Cargill must support multiple languages, cultural differences, and so on; centralizing any part of the operation places unique demands on the company.

When Loren put out a general e-mail in his company asking for multi-departmental feedback on Open Source, he was bombarded with responses. Based on that, he said he felt reasonably confident that his remarks appropriately represented perceptions and judgments of IT people in the company at large.

Cargill is known for its skills in managing risk. The company will look at the state of the market along with the risk of supporting global applications on Open Source before any large rollout. That means, of course, that the company is typically not at the leading edge when it comes to technology. Before any adoption of Open Source would occur, questions of global availability, global internal and external support, and cost must have solid answers.

So is Open Source acceptable to Cargill? The Open Source people at the company told Loren, "Absolutely, yes!" Business management people offer a less enthusiastic reaction. They have heard of Apache. They acknowledge that the company does Perl development, but probably without realizing it is Open Source. They would also recognize the names Tomcat, Samba, and Linux, the latter two of which are under evaluation by Cargill. The company also uses a number of Open Source development tools: CVS, Bugzilla, Axis, Log4j, Struts, ANT, OpenSSH, OpenSSL, JBoss, Jakarta tools, Xerces, and Xalan.

From Cargill's perspective, there are six big issues that must be resolved before the company goes much further with Open Source. At the top, Loren put education to ensure that management understands the benefits of Open Source. As he indicated earlier, they probably don't even know that Perl is Open Source, and Linux doesn't have an Open Source patina because IBM and HP support it.

The other five issues are mind share in the market (to Cargill management that means that more approved vendors support things like Linux), version control, the fact that there are so many players in the space of varying degrees of stability, licensing, and legal concerns.

Cargill will find Open Source more acceptable when applications become more mainstream; market share increases and, as a corollary, the reluctance of using Open Source will then decrease; and when management has been educated properly about the value and risks.

**Stormy Peters, of Hewlett-Packard's Open Source Program Office** next made the case for Open Source acceptability in business. HP is a leading global provider of products, technologies, solutions and services to consumers and business. The company's offerings span IT infrastructure, personal computing and access devices, global services, and imaging and printing. This company of 140,000 employees with capabilities in 160 countries and doing business in 15 languages uses Open Source throughout the company—in IT, R&D, and other places internally—and ships Open Source solutions externally.

The proliferation of Open Source use within the company provoked HP to create an Open Source Program Office; initially, it seemed to come out of concern over “contamination” of sorts. The Program Office took shape with three major parts:

- The first part focuses on Strategy and Policy: What is HP trying to do with Open Source? Why is the company going to use it and ship it? Bruce Perens, who was a consultant to HP at the time, drafted the first version of the strategy and policy document, which is online, and explains to all HP employees the basic rules of use. If they decide to use Open Source, what are the considerations? If they work on Open Source at night, can they use HP machines, and if so, how?
- The company also has an Open Source Review Board to review every instance of Open Source software that HP ships. This is a huge task, given that two to fifteen new projects surface every week that use Open Source software or involve shipping it as part of an HP package.
- The third part is the external presence, including the HP web presence that interactively engages people and participation in conferences, such as this one.

In addressing “is Open Source acceptable in business,” Stormy answered firmly that it is, and cited the billions of dollars generated each year by Open Source products such as Linux and Sendmail. She also paraphrased a quote from HP CEO Carly Fiorina, who has addressed many Open Source conferences over the past few years: It’s not a matter of questioning whether Linux is here to stay, it’s a matter of asking which part of the world is Linux going to dominate.

Given that Open Source is here to stay, the issue then becomes how to make it work effectively, including how do companies make money on it legally. It doesn’t fit into the comfortable paradigm of shrink-wrapped software from a known source, with known support, and known results. To know “what are you really buying?” means being savvy in at least six areas:

- Copying – Customers can freely copy Open Source software, however, vendors may put service agreements on top of a package that impose limitations on copying.
- Support – It can come from multiple places, including doing it yourself, but the way a customer acquires the software may well determine what kind of support is readily available
- Media/Manuals – Documentation is one area in which the Open Source community needs help.
- Licenses – What is a GPL license? What are the differences between the 40+ available licenses?

- Bundles – When a customer buys Linux, for example, what’s really in it? Downloading two different versions of Linux provides two different things; it’s important to differentiate between them.
- Indemnification – The Open Source community does not have a standard system of warranties and indemnification as mainstream IT vendors do. Accountability for fixing a problem may not fall to anyone but the customer.

Stormy’s enthusiasm for Open Source software was unmistakable in proffering how and when it makes business sense. Central to this discussion was the integration of Open Source into new business processes: Absent a range of services from a major vendor like HP, a company needs to look at the testing and certification that may be involved in getting a good fit. Another point of emphasis was licensing. In short, she advised talking with an attorney.

**Carolyn A. Kahn of The MITRE Corporation** is working on a major research project on the business case for Open Source software. MITRE is a not-for-profit organization chartered to work in the public interest; most of its sponsors are government entities. As a national resource, MITRE applies expertise in systems engineering, information technology, operational concepts, and enterprise modernization to address sponsors’ critical needs. MITRE manages three Federally Funded Research and Development Centers (FFRDCs): one for the Department of Defense (known as the DOD Command, Control, Communications and Intelligence FFRDC), one for the Federal Aviation Administration (the Center for Advanced Aviation Systems Development), and one for the Internal Revenue Service (the Center for Enterprise Modernization). MITRE also has its own independent research and development program that explores new technologies and new uses of technologies to solve its sponsors’ problems in the near-term and in the future.

MITRE received a Leadership Award from the non-profit Potomac Forum for investigating the technology and economics of Open Source software in its research project “Open Source Software in Military Systems” and the study is available on both the MITRE and The Open Group web sites. The primary conclusion was that, in many instances, Open Source is acceptable as a long-term, viable solution, but there are risks that must be acknowledged and mitigated.

Specific determinants in choosing Open Source software over COTS (Commercial Off-the-Shelf) software are project-based. According to the study, Open Source tends to be a good option for products relevant and interesting to a large community with a lot of developers working on it. Open Source often compares favorably for server and embedded systems that may require some customization. Open Source can provide a lot of advantages for long-lived embedded systems because of its life cycle licensing and support savings. Open Source software, however, generally does not fare better than COTS for typical desktop applications.

Carolyn asserted that Open Source has had a very successful track record as evidenced by Emacs; Apache, which comprises more than 60 percent of the web server market;

Sendmail, which carries nearly 90 percent of e-mail traffic; and Linux, which has an estimated worldwide user base of 18 million. She also cited Open Source processes such as Perl and TCP/IP as evidence of Open Source success.

The study looked at the benefits and risks associated with both Open Source and COTS and the summary results appear in the following table:

### Comparison of OSS to Traditional COTS

Typical Benefits	Typical Issues/Risks
<ul style="list-style-type: none"> <li>▪ Technical excellence, efficiency (fewer lines of code)</li> <li>▪ Rapid release rate of fixes/patches</li> <li>▪ Easy to manage (central administration, remote management)</li> <li>▪ Ability to tailor source code to meet specific needs, tightly control system resources</li> <li>▪ Re-use of code already written by another user</li> <li>▪ Lifetime of OSS systems and their upgrades can be extended indefinitely</li> <li>▪ High degree of interoperability</li> <li>▪ High quality support at minimal costs (competitive)</li> </ul>	<ul style="list-style-type: none"> <li>▪ Poor code if OSS project is small and attracts interest of few trained developers</li> <li>▪ OSS process has tendency to focus on technical user at expense of non-technical user</li> <li>▪ Need for version control if system requires integration and development</li> <li>▪ Risk of fragmentation</li> <li>▪ Lack of available applications</li> <li>▪ Seen as competitor by comparable or substitute products</li> </ul>

Carolyn’s summary advises that, to assess the feasibility to the Program Manager, both the economic benefits and costs of Open Source usage and maintenance must be evaluated over the full life-cycle. She then recommended ways of doing so. It involves a system of weighted variables in which the Program Manager ranks the following from “very strong” to “very weak”:

- Ability to customize
- Availability/reliability
- Interoperability
- Scalability
- Design flexibility
- Lifetime
- Performance
- Quality of service and support
- Security
- Level of difficulty/ease of management

- Risk of fragmentation
- Availability of applications

Carolyn also examined the ways to assess cost, including the direct costs of hardware and software, staffing, the internal and external support costs, and de-installation. In addition, she noted the indirect costs of support—mostly training—and downtime.

In looking at the “Buy versus Build” option, a couple of key points surfaced beginning with the assertion that pure COTS can be analogous to an unmodified Open Source scenario. In both cases, it’s important to assess the reliability and functionality of the product, licensing restrictions, and so on. On the build side, the costs of acquisition and support must be considered.

Carolyn concluded with five steps to determining whether to choose Open Source Software or COTS, as follows:

1. Assess supporting OSS developer community (e.g., Linux)
  - Look for large, talented, and well-organized communities
2. Examine the market
  - Is there strong and increasing demand for the OSS?
  - Have complementary services emerged in the marketplace to provide needed support not available from the community?
3. Conduct a specific analysis of benefits and risks
  - OSS taxonomy of benefits and risks compares products relative to specific economic/performance/mission objectives
4. Compare the long-term costs
  - OSS cost element taxonomy compares long-term costs associated with usage and maintenance relative to objectives
5. Choose and execute your strategy
  - Steps will provide information/detail to choose and then execute the most effective option combination of OSS, traditional COTS, and proprietary development to support objectives

**Bob Vavra of Unisys** launched the Question & Answer session following the panel by asking Andrew how engaging with the Open Source community is different from engaging with a typical vendor.

Andrew responded that there are two aspects of the Open Source community. It is an amalgam of developers from across the globe, but it is also those commercial enterprises that support Open Source applications. Deal with them in slightly different ways, he advised.

With the community, after a company decides which application it wants to deploy, it should start building a relationship with the developers. A large enterprise should designate someone to be primarily responsible for that relationship. Acknowledging people like Bruce Perens and John Terpstra in the room—both of whom are Open Source



pioneers—he emphasized that it is important not only to use output from the Open Source community, but also to contribute something back to it.

The advantage of working closely with the Open Source community will surface in the form of additional functionality for the application, for example. But that will not happen, he cautioned, unless they see some reciprocity. That could take the form of contributing code or bug fixes, for example.

Andrew told a story of doing this in relation to developing a time-card tracking module related to a time-management application. His company contributed it and, in return, asked for help with three additional features to the project management system. Optimistically, they also threw in a wish list that included a few others that the community could be thinking about at some point down the road. Within 48 hours, 25 new sets of functionality were written into it!

Also in response to Bob Vavra of Unisys, Bruce Perens noted that Unisys is a big company with a lot of intellectual property. He urged caution about interacting with the Open Source developers so that management doesn't panic because they fear contamination of the company's intellectual property. The risk occurs both when a company uses Open Source code in shipping products, and when the company makes contributions back to the community. Without question, a company engaged in these activities should have an internal Open Source policy.

He also noted that software is a cost center. A company will pay a certain amount to outsiders for proprietary software. Bruce asked the audience to consider what software is worth to them and what they should be putting into Open Source development to encourage it. Even with a contribution of money or resources, the cost of Open Source software is probably significantly less than the alternative.

**Hank Jones, speaking as the former VP for a big company like Unisys,** suggested that it is good to advocate changes in Human Resources processes in the area of performance evaluation. Developers, product managers, marketers, technical support people or others who are expanding the company's capabilities through their work in Open Source should have that acknowledged in their formal reviews.

**George Behnke of Cray** posed interrelated questions about how to make sure that the Linux kernel won't fork or fracture and how to standardize Open Source. HP's Stormy Peters referenced the Linux Standard Base produced by the Free Standards Group, conformance to which is certified by The Open Group. It is a start, but a major barrier to further standards efforts in Open Source is the community's lack of appreciation for standards. Members of the community are not engaged in the effort; they haven't been convinced of the value.

**Eduardo Gutentag of Sun Microsystems** added that it is important not to confuse Linux with Open Source. Linux is Open Source, but Open Source refers to far more than

Linux. There may be different strategies, therefore, in introducing standardization to the Open Source community.

Addressing the first part of George's question, **Chris Hertel of the University of Minnesota**, who contributes to the Samba project, noted that there is a commercial aspect that will prevent the forking of the kernel. Large OEMs make it clear that they do not want to go through anything like the UNIX® system wars again. The issue is that in order to drive the adoption of Linux and other Open Source applications there have to be more software vendors that are enabled on Linux, and if they had multiple kernel choices, that would make the challenge so much more difficult.

Bruce Perens added that, in the Open Source community, forks are often considered hostile activity, and that licensing actually helps "heal" forks.

Graham Bird summarized much of the discussion by saying that, the general advice seemed to be to buy certified products, otherwise there was no guarantee of what's "in the box."

Introducing the topic of return on investment, **John Schmidt of Best Buy** asked if there are any definitive studies that identify sweet spots—solid ways to demonstrate ROI.

Carolyn Kahn noted that there are many studies available, and many of them advocate different points of view. After looking at them, she concluded they are not reliable in, or applicable to, every setting.

Andrew Aitkin suggested taking different pieces of the studies and building one's own model. Certainly, there will be fuzzy areas, such as developing cost factors around Open Source models, but the studies provide guidance.

Loren Sinning added that a company must look at its business requirements first—what is the perceived business value of going with Open Source? Definitely don't stop with a look at what the software costs versus what functionally similar proprietary software might cost. Support, maintenance, and other follow-on costs must be factored in. On the positive side, don't forget to consider the advantage of flexibility that Open Source allows, of being able to port an application to any hardware architecture desired. He asserted that that flexibility is where the ROI is.

Chris Hertel reminded everyone that one of the advantages of using Open Source is that it provides access to the source code. Internal company developers can look at and become involved with it. If people on staff do that, the company's investment changes. Suddenly, there is a human investment in the products the company is using; it becomes a more exciting environment for employees. It can provoke a cultural shift.

The variations on the "advantage of flexibility" discussion really come down to control issues, according to Bruce Perens. Companies want control and feel they have it, even if it's an outside source that modifies the product according to their needs as opposed to

someone on staff. It's the fact that they can do it that provides the satisfaction. It removes fear about another company being able to block one's business.

**Andras Szakal of IBM** felt strongly that Chris' assertion wasn't valid outside the research environment. Companies don't have the luxury to probe code and play around with it; they have a core business that requires their expertise.

A major discussion about the upper limits of Open Source ensued when **Greg Wettstein, systems architect for North Dakota University** and an Open Source pioneer, asserted that the many Open Source developers who do infrastructure software have little or no interest in developing the types of applications that most people in the room want to deploy in their organizations. That will be the glass ceiling that Open Source hits.

**John Terpstra, founder of the Samba team**, countered with a unique point of view, since his Samba initiative reflects a desire to capture a significant slice of the Windows networking market. He also had the specific intent to keep as low as possible beneath the radar that perceives this Open Source activity as a corporate threat. Questioning what makes a successful Open Source project, John suggested that the thought behind it is that there are patterns that might indicate what Open Source projects will be successful in the future.

He also took on Bruce Perens and his allegation that the central issue is control. John felt that the issue is not one of control, but of risk management. There is a strong perception that if one buys a commercial package, there is protection. Open Source, however, is written by "bearded monsters who wear sandals because they can't tie their shoelaces." Therefore, he said, that constitutes a big perceived risk.

John told the story of getting involved in Open Source. He was involved with a company called Overseas Periodicals, operating between Europe and the Southeast Asian perimeter. Having invested \$5 million US to develop an application on which the business depended, they went through extreme due diligence to ensure they had the entire source code. Unfortunately, the company they contracted with to provide this application never gave them one vital tool to make sense of it all—the development system it was built in. When that contractor was gobbled up and the acquiring company disposed of all the people associated with the Overseas Periodicals project, there was no one to support the application. John emphasized that this story underlines the value of Open Source in mitigating risk.

Andrew Aitken furthered the "glass ceiling" discussion about the problem with producing Open Source applications for business by saying that outsourcing applications is only part of the story. There is also what he calls insourcing, or migrating the application development to an in-house Open Source development team, and he cited a specific example within the Federal government that evidenced its cost effectiveness: Not in year one, but within a three-to-five-year period, the cost savings were enormous.

Once again, Bruce Perens sparked some different thinking about the glass ceiling with a reminder that, a few years ago, the limit was set at the GUI. Now it is set at enterprise applications. In truth, where that ceiling is set is an unknown.

Greg Wettstein countered by wrapping up his thoughts on why he thinks the glass ceiling concept is valid and what can be done to break through that upper limit. In his opinion, the economic model that underlies Open Source will drive change. Companies must support the (Open Source) companies that are trying to create the enterprise applications. If organizations embrace a model where they simply take Open Source software for free and provide no form of remuneration for the people who are investing their intellectual property in it, there will be the glass ceiling. Alternatively, companies will just not have access to the level of high quality software that they can reliably bet their organizations on.

Andras Szakal reinforced that thought by noting that large companies such as IBM and HP invest relatively heavily in testing, quality assurances and support for Open Source products because their enterprise customers demand it. That is the only way to make Open Source acceptable to business customers.

Graham Bird concluded the glass ceiling discussion with the assertion, “If Open Source is going to succeed, it has got to come with the mindset of the customer, and in many instances that doesn’t happen.”

He also prompted the audience to indicate their degree of interest in The Open Group supporting the development of a generic ROI model for Open Source. Some audience members did, indeed, express interest.

The session ended with a topic proposed by Hank Jones who wondered how many organizations in the room had multi-disciplinary teams devoted to Open Source projects and, more basically, how many had policies in place for Open Source. The result of the latter was very few, with even fewer having multi-disciplinary teams in place.

### ***The Technical Panel***

*Moderator: Terry Blevins, Vice President and CIO, The Open Group*

*Panelists:*

*Bruce Perens, Perens LLC*

*Dr. John Collins, Department of Computer Science, University of Minnesota*

*John Terpstra, Samba*

*John Schmidt, Best Buy*

*Eduardo Gutentag, Sun Microsystems*

*Andras Szakal, Federal Software Group, IBM*

Terry Blevins opened by expressing his concerns about whether or not Open Source is ready for small to mid-sized enterprises (SMEs). For SMEs, particularly not-for-profits,

a restricted budget for IT doesn't allow for a lot of experimentation—for exploring the potential advantages of Open Source.

He then posed the primary questions of the session:

- Is Open Source robust, scalable, portable, and interoperable enough to support an enterprise?
- Is Open Source usable for SMEs?
- What Open Source technologies are coming down the road, such as infrastructure and applications?

Corollaries to these would be questions concerning what SMEs have to do to exploit Open Source while minimizing risk, understanding the support needs and impact on managing an enterprise environment, and grasping how Open Source can help achieve Boundaryless Information Flow, as described by Allen Brown in his keynote. Terry stressed that open standards underlie the success of this effort.

**Bruce Perens, Open Source consultant and pioneer**, tackled the first question with a question: Is robustness always a goal of software? Maybe not. Clearly addressing his next question to the vendors in the audience, he asked if they really want their software to always be robust, easy to use, and trouble-free—which would negate the need for support calls. He wasn't being entirely facetious and used this idea as a provocative way of introducing a significant reason why an enterprise would turn to Open Source—an entire community is available to improve the robustness of a product.

He provided an instance of a very robust Open Source product from the Open Source Developer Lab, which is producing “carrier grade” software. Designed for use in a telephone company central office, he stressed that failure could not be tolerated in that environment. In short, Open Source can yield robustness.

On the issue of scalability, he saw no real problems. Regarding portability, he asserted that Open Source scores highest; the operating system runs everywhere and so do its applications.

As Bruce sees it, Open Source shares a lot of goals with SMEs. To start, the biggest challenge they face is competing and surviving in a field of huge, multi-nationals; Open Source has a lot of the same issues. Open Source has an SME-oriented, user oriented mindset. The big thing that SMEs do need is better commercial support.

The next thing coming down the road is Open Source on the desktop. Open Source products such as OpenOffice and Mozilla perform the entire range of functions required by most clerical employees.

And what should SMEs do to minimize risk? Responding to this, Bruce made two key points. First, he returned to one of the major pieces of advice that emerged from the business panel: Have an Open Source policy in place, particularly if the company has intellectual property concerns. He also cautioned them to understand the support ramifications. Red Hat, for example, has its own support contracts that depart somewhat

from the norm in the Open Source world, so an SME needs to pay attention on a case-by-case basis. In short, make sure there are a number of viable support options for the software.

Regarding the issue of managing the impact on an enterprise environment, Bruce asserted that management doesn't usually know how widespread the use of Open Source is an organization. The first step is knowing exactly what software resides in the company.

In no uncertain terms, Bruce noted that Open Source has a central role in achieving Boundaryless Information Flow. He referred to the lock-down of systems with increased digital rights management and deployment of trusted systems, and expressed confidence that Open Source provides the option for releasing the flow of information appropriately. Because Open Source is able to implement trusted systems, and secure use of data, it provides security while allowing the customer, and not the software, to be in control of whether that happens or not.

**John Collins, University of Minnesota professor**, draws much of his perspective from the 30 years he spent in industry prior to entering academia. His exposure to an Open Source approach began very early with work on a DECUS Pascal compiler. Currently, he works in a nearly Microsoft free environment.

First, he concurred with Bruce Perens that the answer to the first question (robustness, scalability, etc.) is "yes" on all counts, and he agreed that it's true to varying degrees.

On the second question of Open Source's usability for SMEs, he disagreed with Bruce and contended that the desktop applications are not nearly as good as they need to be. The pieces, such as OpenOffice and Mozilla, do not work together. The need to share information easily is a big reason why Microsoft dominates the desktop arena; it offers usability and interoperability. Open Source infrastructure software is interoperable at the infrastructure level, but Open Source desktop software is not as interoperable as needed between desktop applications.

Moving to areas of success as well as challenge in enterprise adoption of Open Source, John said that one of the reasons that government agencies are starting to adopt it is because of the document formats. He sees an opportunity to develop document formats that are common and well documented and have implemented applications around them to drive this process forward.

**John Terpstra of the Samba Team**, one of the people who brought TurboLinux into the US marketplace, has been part of the Open Source movement since 1981. Samba is an Open Source/Free Software suite that provides seamless file and print services to SMB/CIFS clients. Samba is freely available under the GNU General Public License.

John quickly addressed whether or not Open Source Software is robust by citing Linux and xBSD as examples of mature and robust Open Source offerings. This came with a footnote, however. He said that Open Source has successfully challenged Microsoft's

business model, and as a result, Microsoft has made strategic initiatives with Windows Server 2003, which he freely admitted is a great improvement over Microsoft's previous offering. He added that it will challenge perceptions of Open Source in some areas.

He progressed by asking the audience how many Apache servers they thought were in the market. The answer: Sixty-two percent of servers that power the Internet are Apache servers. He then quizzed the audience about how many Samba servers were in the market. Independent research done in 2001 indicated that the Samba install base was between 10 and 14 million servers. That doesn't quite jibe with reality as presented by other researchers, however. IDC statistics put the total number of servers at 25 million, ten percent run Linux; ten percent run UNIX. The rest are 3.8 million instances of NetWare and 16.2 million instances of Microsoft platforms.

His next question to the audience: What do these numbers—at odds, yet all indicating a large installed base of Open Source products—tell us about the challenges, especially in light of Microsoft's new high performance server?

In the areas of applications, interoperability, and portability, John assigned a “good enough” to each category.

Open Source seems to be making inroads in enterprise use. According to a June 15, 2003 *SD Times* article, between 36 and 38 percent of all corporate sites run Linux. John attributed this to its speed, ease of installation, maturity, and the fact that it works well. The two-fold problem, on the other hand, is configuration and manageability. Setting up a DNS or DHCP is difficult to do. This problem is the Achilles heel of Open Source, according to John.

He then explored the new version of Samba, to be released within the next 30-60 days of the conference and challenged the audience to tell the Samba team what they need, what they find that's good and bad. Techies, he said, focus more on the neat things they can do rather than the requirements of the enterprise.

In preparing for the new release, the team found that most of the documentation they had written was slanted toward the UNIX administrator, toward the UNIX user. Ironically, the bulk of Samba's opportunity is in the world of Windows. As a consequence, they went back and repositioned it. The hope is that their continued awareness of Microsoft's dominance, success and shortcomings will help them meet the demand for alternatives to Microsoft Active Directory and Microsoft Exchange Server, as well as for better configuration tools.

John concluded by contending that the key obstacle for Open Source is the business model. The Open Source community needs to determine where the money is and get it coming in to allow development of products that will make it more successful. He placed importance on the development of tools that will provide end-users with more freedom in what they will use.

**John Schmidt, IS Leader for Best Buy**, stepped up next to address the same set of questions from the perspective of a retail organization of 550 stores across the United States and 150 in Canada.

John opened with a look at the integration spectrum at Best Buy. He described both integrations systems and their framework, beginning with a view of the integration hubs—messaging hubs, ET&L (extract transformer and load) systems, external integration hubs, file replication hubs, and web services hubs. An integration competency group within the organization designs, builds and sustains all of the integration system and is composed of about 35 employees and 100 contractors, 40 percent of whom are off-shore.

The metrics he presented focused attention on the magnitude of his IS challenge. Best Buy moves about 100 Gigabytes of data a day between systems, and there are 350 different systems (applications) within the company.

Best Buy also has something it calls an Integration Factory. This is where the analysis and data gathering pertaining to integration occurs; the offshore factory then uses the specifications to design, assemble and test a fully functioning adapter from reusable building block components. Using this framework, Best Buy has built over 500 adapters, 400 in the period June 2002 and 2003. Overnight turn-around is not uncommon. He then quantified the value of frameworks. Highlights included cutting real-time interface cost by more than two-thirds, from \$20,000 to \$4,500 and the cycle time from 20-30 days to four-six days.

John categorized Best Buy as a user of Open Source and pointed to twelve different products in use, including Eclipse, Apache web server, Linux, and SendMail.

He then went straight to the standards discussion, asking “What’s wrong with this picture?” in presenting a graphic depiction of how few end users participate in standards activities related to the retail sector. Those who provide solutions to the sector dominate the standardization process because users have abdicated their responsibility—not because they can’t participate.

He then took a position on what he sees as the differences between traditional standards and Open Source. Primary contrasts are as follows:

	<b>Traditional</b>	<b>Open Source</b>
Driver	Vendors	Users/Developers
Process	Analytical	Empirical
Acceptance	Agreement	Evolutionary use
Deliverable	Specification	Software (Code)
Motivation	Marketing/Sales	Operations (making it work)

The next panelist was **Eduardo Gutentag, XML Standardization Lead, Web Technologies and Standards, Sun Microsystems**. Acknowledging that Open Source



offers many positives on an IT balance sheet, he chose to focus on the obstacles to acceptance. Key among them are:

- Developers' lack of experience with enterprise
  - Lack of familiarity with requirements and acceptable solutions
- Developers' lack of testing facilities for interoperability and testing
- Developers' perceptions
  - Testing is boring
  - Bug fixing isn't as creative as developing something new

Eduardo mentioned that developers' being out of touch with the needs of the enterprise is not the exclusive domain of Open Source. It happens whenever developers collect to create and users are not involved.

Regarding the usability of Open Source for SMEs, he maintained that one should always make the assumption that SMEs don't have a lot of resources for IT. They need reasonably priced support and reasonably readable documentation for basic functions such as installation.

He looked down the road at opportunities for Open Source and saw that almost every functionality may be open sourced very soon; the real money to be made is in support. In asserting that, he openly disagreed with some expressed points of view from other speakers and audience members. He also cautioned that there is too much flawed product in the marketplace.

One way to address the quality—or at least the consistency—problem is through open standards. Eduardo provided his definition of open standards and differentiated, as Allen Brown did in his keynote, between standards as products of recognized standards-setting organizations and specifications. The primary features of open standards in Eduardo's definition are:

- Specifications that are reliable
- Free from the threat of legal encumbrances
- Able to work across development and deployment environments
- Subject to peer review and input throughout their lifecycle

He stressed the urgency of having such standards in place as we move forward in Open Source, and called on The Open Group to start an education program about standards.

### **Andras Szakal, Chief Architect of the Federal Software Group for IBM**

In foreshadowing the contents of his presentation, he noted that he would not only talk about why Open Source is important to IBM, but also how IBM positions Open Source software as part of its offerings.

First, he reminded everyone of IBM's early, traumatic encounter with desktop computing. He squarely blamed a lack of listening to customers as the cause of a revenue plunge to less than half of what it had been—in about a six-month period—that necessitated a corporate resurrection. He credited Lou Gerstner with IBM's resurrection. As a vendor, being inwardly focused is not a formula for success, he summarized. Everything should be focused outwardly on the end-user.

As a company, they realized that products needed to be focused on open standards, as well as on Open Source when it grew in popularity. The trend occurred when money and people moved towards it, a community and standards came into play, and customers began demanding it. Of course, some key vendors resisted Open Source throughout this shift.

IBM considers Open Source software important for three big reasons: Customers want it, Open Source software is a good approach to developing open standards, and it can be a source of industry innovation. In a sense, vendors use the Open Source community to create a base of knowledge and extend their ability to develop software.

IBM's goals for Open Source begin with the drive toward rapid adoption of open standards. They want to cooperate on standards development and compete on the implementation of standards. IBM also wants to use Open Source software as a business tool to block competitors from creating “lock-in” and proprietary control points. Finally, the company wants to extend IBM mindshare, to create a preference for IBM by linking to popular Open Source products and projects.

In his look at the road ahead, Andras called attention solely to the projects of greatest importance to IBM, namely, Linux, Eclipse, Apache, and Grid Computing. He asserted that Grid Computing will change the whole IT infrastructure and is an open standards, Open Source initiative from the ground up.

In positioning Open Source software within IBM's strategy during his closing remarks, Andras offered eleven areas of consideration, beginning with product support. As other speakers had said before him, commercial vendors tend to offer well-defined support for enterprise customers. Other areas he cited were platform support, long-term viability, scalability, reliability, integrated tooling, support for open and government standards, dependence on individuals, security, perceived TCO, and partnership.

Terry Blevins kicked off the Q&A with the provocative question: “From a technical perspective alone, what was wrong with UNIX that Linux came into being?” Bruce Perens succinctly stated that there was nothing wrong with UNIX from a technical perspective. Eduardo Gutentag added that the popularity of Linux—the perception that it's “in”—has nothing to do with the technicality of UNIX or Linux. He suggested that it's matter of social engineering. Andras Szakal, however, noted that UNIX was often slow.

Bruce Perens came back with an assessment of what went awry for UNIX. There would have been a lower cost, and higher performance and quality if there were good competition, he suggested. He talked about the profit-focused motives of the UNIX vendors and the lack of motivation to improve the product. The difference with Linux and Open Source is that the market dynamics are different for the Open Source community, which does not have the same objective as the UNIX vendor. He concluded that it is that difference that pumps adrenaline into an environment that had been boring a few years ago.

Discussion turned intensely to open standards when someone from the floor asked Eduardo about Sun's handling of Java standards, and the perceived move of the company to keep Java proprietary. Eduardo maintained that the Java Community Process (JCP) is open, but several audience members joined with the original questioner in asserting that the costs of participation and product certification are not elements of an open process. Eduardo noted how expensive it is to participate in other open standards arenas and how expensive it is, in many instances, to earn certification for a product.

Bruce Perens intervened by saying that, by Eduardo's own definition, Java appeared to meet the criteria for an industry specification rather than an open standard. Eduardo maintained his position.

As the conversation developed around the open standards issue, a key question emerged: Who has the controlling interest? With proprietary software, the vendor decides what goes into the package, even though that decision might reflect user input. With Open Source, is that somehow also the case, or does it genuinely reflect a more open process of development? Bruce Perens stated emphatically that Linus Torvalds only exercises ultimate authority over the version of Linux that his organization distributes. Anyone else could use a Linux kernel and, in compliance with the GPL license, do whatever he or she wants to do, including distribute it widely. Java, however, is protected so that an analogous development could not occur legally.

JBoss is the leading implementation of J2EE, Bruce said, and—despite contrary opinions from the audience—went on to assert confidently that it has more users than any other version. JBoss cannot be certified as J2EE, however.

As general counsel for JBoss, Larry Rosen of Rosenlaw.com insisted that that was not exactly true.

Graham Bird introduced the thought that testing for conformance to standards is expensive. It was, indeed, a potential barrier to Open Source adoption, but the fact is that people have to be paid to develop reliable test suites. It is currently a legitimate business expense, but it would be ideal if the community could find a way out of the high-cost model. Chris Hertel of Samba noted that Samba does come with test suites that are free. They are not standardized, but they are the tests developed along the way to ascertain how good Samba is.

Terry pulled the conversation back to a clean focus on open standards rather than tangential conversations about Java and the expenses related to Java compliance. Chris asked for guidance about participating in standards process given his staff situation at a university.

Multiple responses supported the notion that anyone, even individuals, could participate with little or no hard costs in the open standards process. From certain standards (or specification) development groups that allow online participation to actual meeting attendance with voting rights, the opportunities run the gamut.

**Amy Marasco, General Counsel of the American National Standards Institute (ANSI)**, highlighted the different points of view that define “open” in open standards. There are the process, legal, and technical points of view. The process issue involves the ability of all stakeholders to participate with voting rights, or at least the opportunity to render an opinion through public comment. This is how the ANSI process works. Someone with legal issues at the forefront might define “open” in terms of intellectual property. They might consider a standard “open” only if it is royalty-free, and unencumbered by patents or other IPR claims. Technically, an open standard would allow for the unrestrained exchange and consideration of technical information in developing a standard.

**Bill Estrem, a co-host of the event from the University of St. Thomas**, interjected the idea that “free” is not necessarily in anyone’s best interest, that is to say, that if customers want vendors that are sustainable, they must be able to make a dollar.

**Elizabeth Rough of Unisys** shifted conversation to the critical point of users of Open Source contributing code back to the community to add value to Open Source products. Bruce Perens responded that, in terms of obligation to share any modifications of software, there is none if the party is simply running software under the GPL license. The obligation rises with distribution, and it is to distribute the source code to those whom they distribute the binary and not restrict those people from distributing it. His summary advice was to work out business goals before choosing the Open Source license so that it’s clear to what extent the company is obligated to share.

Greg Wettstein, whom Bruce Perens singled out as a key person on the early Open Source effort, moved the discussion to focus more on risk mitigation from a technical perspective. He referred back to 1997 when he was part of a group to do a build-out of the Linux architecture at North Dakota State University. The risk assessment team asked itself what significant risks were they exposing the organization to by platforming on top of Linux and Open Source. Greg’s team identified the primary risk as Active Directory. Potentially, AD would have a huge impact across the application base due to its ability to control the identities of the people in the organization, as well as what they were authorized to receive. To fix that problem, they designed a replacement for Active Directory and created a unique authorization technology. Greg’s business counterpart immediately patented that technology. His question centered on his being an Open Source advocate with a unique piece of technology that it potentially critical to the

acceptance of Open Source. The dilemma is that it is unique and that major vendors will realize that it is valuable and something they wish to acquire and market with value added services. If a member of the Open Source community produces something like this and patents it, he wondered, would the Open Source community accept it and help them develop it?

He clarified that the question was not related to patents. It was more a question of how and whether innovation can prosper in the Open Source environment.

Bruce Perens responded that, in his view, it was a question related to patents. He posed questions about who funded the research for the innovative technology, whether or not the terms of the patent allowed free access, and nuances about the definition of a derived work. He followed up with a recounting of recent discussions at a World Wide Web Consortium (W3C) meeting at which the question arose: If you have a patent with royalty-free terms, what is the point of having a patent? The conclusion they arrived at was that such a patent is for the purpose of implementing a royalty-free standard and for no other purpose. If one tries to use that patent in a different way in the same program that implements the patent, then a license would be required. He also noted that this approach is probably not compatible with the GPL license.

### ***The Social and Ethical Panel***

*Moderator: Bill Estrem, Professor, College of Business, University of St. Thomas*

*Panelists:*

*Dr. Ken Goodpaster, University of St. Thomas*

*Dr. Peter Vaill, University of St. Thomas*

*Malcolm Reid, Medtronic*

*Tony Stanco, The Center for Open Source & Government, and Cyber Security Policy and Research Institute, The George Washington University*

Bill Estrem, a member of The Open Group's Governing Board, prepared the audience to "plow new ground" by exploring the social and ethical dimensions of Open Source and open standards. He explained that some people he approached did not feel qualified to address this topic and called the current panelists "brave." He noted that, despite his colleagues Ken Goodpaster and Peter Vaill not being at all technical, they have often come into his business classes to address social and ethical issues related to e-commerce, for example.

Further differentiating this panel from the others, Bill described his group as facilitators of discussion about the overarching topic, rather than experts about any specifics of Open Source. He set the tone for the conversation by pulling out a memory from his corporate days. At the time, one of his projects involved the OSI networking model, which had seven layers. "There were two additional layers not revealed in the public interfaces. The eight layer was politics, and the ninth, religion." He then posed the primary questions about open standards:

- Who benefits from standards?

- What motivates participation in standards efforts?
- What is the value of social capital in the current economic milieu?
- How do we deal with the dual tragedies of
  - Tragedy of the Commons
  - Tragedy of the Anticommons?

Explaining the latter, Bill harkened back to village controversies over access to common property, or “the commons.” If there were no fences, then the animals could graze there alongside the people who were enjoying the commons, and that would ultimately destroy it. Fences preserved the commons, but kept out a lot of people. The current conflict over royalty-free versus RAND, he felt, are analogous; that is, whether open standards should be unencumbered by IPR that embed cost in the use of the standards, or whether Reasonable And Non-Discriminatory use agreements may be acceptable.

Bill moved to topics for the Open Source discussion, first considering the human organization aspect of Open Source. “When we look at the human organization, we see something that is far more difficult to manage than the diverse technical elements.” He posited that doing technology development and innovation in a hierarchical environment is hard enough, but coordinating such an effort in a virtual community with people of disparate cultures who don’t know each other is a daunting challenge. And how can companies of different sizes that want to enter that community and contribute something go about it effectively, he wondered.

Before turning the floor over to the panel, Bill asked an important question: “Is what we are doing with Open Source applicable to other activities?”

**Dr. Ken Goodpaster is a moral philosopher on the faculty of University of St. Thomas**, and formerly a faculty member at Harvard’s Business School. Moral philosophers and ethics experts rely on the “moral point of view” and Golden Rule in discussing issues, and Ken suggested that they might be applicable in this technical arena of an Open Source conference.

He cautioned that ethics should not be confused with altruism. Ethics is not about altruism; it is about the pursuit of self-interest. Ken also said that ethics is about being partial, and then breaking open that partiality and generalizing it to others. In chiding him for some selfish behavior, a friend of Ken’s once told him, “You’re special, but you’re no damn different!” That optimizes the essence of ethics and the moral point of view. How is it possible to live one’s life with the awareness that one is special and unique, and that one is no different at the same time? He said it’s not just a personal problem, but also an organizational one.

He focused on the clusters of theories of “logics” that have endured over time—the four forms of sorting out relationships in terms of moral reasoning:

- Interest-based – This is a cost-benefit analysis centered on determining the greatest good for the greatest number of people. The individual says, “I count for one, and everyone else counts for one.” The individual then does a cost-benefit analysis related to behavioral choices and whichever choices come out with the

- highest benefit-to-cost ratio is the right thing to do. Also known as utilitarianism, choices that create pleasure and generate good will, for example, go on one side of the ledger, whereas those that create pain go on the other.
- Rights-based – The rights-based thinker has claims anchored in nature, not convention. Someone of this ilk would say to the interest-based thinker: “Do you really want to subject to the constant calculus of cost and benefits certain fundamental rights such as the liberty to be free, to worship unencumbered?” Rights-based thinking gave the United States the Bill of Rights.
  - Duty-based – The duty-based thinker would criticize the first two for being too focused on the micro-relationships between people. This kind of ethicist would emphasize communities and obligations anchored in fidelity to relationships. The logic of John F. Kennedy’s famous “Ask not…” rhetoric from his inaugural address is grounded in duty-based thinking. It involves subordinating personal interests and rights to something larger.
  - Virtue-based – A virtue-based thinker would accuse the others of making ethics unnecessarily complicated. The emphasis here is on habits or character traits that guide choices. It is consistent with St. Augustine’s advice to “love and do what you will.” As Allen did in his keynote, Ken called everyone’s attention to the frescoes on the ceiling of the entrance hall that depict the seven virtues of faith, hope, charity, prudence, justice, temperance, and courage—the lynchpin virtues of Western tradition. There is no calculus involved in this approach.

Ken then suggested that the group might look at the moral credentials of the two paradigms of “Open Source” and “proprietary.” He proffered that the differences might be summarized as follows:

<b><i>APPLIED TO PERSONS, ORGANIZATIONS, AND SOCIAL SYSTEMS</i></b>	<b><i>Open Source</i></b>	<b><i>Proprietary</i></b>
Interest-based	Common good	Private interests and public interests
Rights-based	Acknowledgement rights	Property rights
Duty-based	Obligations to share and to cite	Fiduciary and stakeholder obligations
Virtue-based	Generosity, honesty, collaboration	Accountability, fairness; prudence

He concluded by saying that he was offering a way of processing a moral debate between two paradigms.

In introducing **Dr. Peter Vaill, also on the St. Thomas faculty**, Bill Estrem acknowledged his expertise in organizational behavior. With his orientation, Peter said that he looks at how a project stands and falls on people behaving effectively toward each

other. The behavioral scientist asks, “Why or how did this happen?” or “What will happen next?” In contrast, the ethicist asks, “Should this be happening?”

Peter opened with a “thought experiment,” posing this question to the audience: What functions and capabilities of an operating system are taken for granted today which were virtually unimaginable 15-20 years ago? Taking this to the next logical question, he suggested that 15-20 years from now, people would look back on today and be able to come up with a plethora of things they once found unimaginable. The question then becomes, how should the industry behave to make possible the emergence of ideals such as the range of “ilities”—interoperability, reliability, scalability, and so on—so that progress does not stall.

His next thought experiment centered on the question: What value systems seem to be in collision regarding the question of Open Source versus proprietary source code? For him the question sparked a dual response.

Paradoxically, there may be relatively few short-term commercial arguments for Open Source. It is natural that an organization would want to protect innovations of its own making: the potential financial leverage of such innovations is very high. The need to recover development costs is continual and intense. There is continuing fear that someone else will create some code that leapfrogs everybody, and it seems foolish to trust that if someone else does create such code, they will readily share it with everyone.

The problem is that what may make sense for the industry may not make sense for individual organizations. Individual organizations experience an imperative to pursue their direct, immediate interests. Organizations develop tacit theories of their survival requirements to justify what appears from the outside to be greed and selfishness.

Next was a strike at the six-sigma mentality. Six sigma impedes innovation by creating performance anxiety, he asserted. Today’s students live in a world where they either perform or they will be fired. The twisted result is a dumbing down of performance. It is a predictable result: Put enough pressure on a human being and it will do what is required, but only what is required. There will be little or no taking chances, of going outside the box. All thought is on the next deliverable, and therefore the immediate project and how it must be produced with zero defects.

Peter next asserted that technical excellence would not carry the day. Directing his remarks to the technical people in the audience, he said that everything they know technically depends for its effectiveness on the meaning that someone else attaches to it. That means they are in the interpersonal world of talking with people, persuading people, sitting around the table in groups and teams—all of the kinds of behavioral things in which technical excellence is expressed. But if they only have the technical excellence, and take a haphazard approach with the interpersonal elements, there is a good chance they will run into difficulties. He focused their attention on the fact that the stakes were high for them. The potential for making money and revolutionizing the industry is ever-present.



Finally, with a nod to John Terpstra's earlier argument that the big problem in Open Source is the business model, Peter offered additional support in his summary presentation on the "balanced scorecard." It is a four-dimensional model of effectiveness of an organization, not just one-dimensional, that is, profit based. In addition to evaluating profit, it also involves the efficiency of work systems, effectiveness with which the company reaches its markets and treats its customers, and the degree to which it is effective with its people.

Next up was **Malcolm Reid, the Director of Technology Architecture for Medtronic**. Probably best known for introducing the pacemaker to healthcare, Medtronic is the world leader in medical technology, providing lifelong solutions for people with chronic disease. Each year, 2.5 million patients benefit from the company's technology, used to treat conditions such as heart disease, neurological disorders, and vascular illnesses.

Malcolm's current dual sets of responsibilities position him uniquely within the neurological and diabetes business sector of his company; it is a joint assignment in business information systems and product development. They are different worlds with common questions. First, quality issues come to the forefront; all products must be safe and efficacious. In fact, striving for "the highest possible quality" is part of the company's mission statement, hence, its value system. It is also a legal and ethical requirement considering that the end-users depend on Medtronic equipment to save their lives. The processes in place for software validation, procurement, engineering, and so on, must withstand intense scrutiny.

Malcolm's sector is now considering an Open Source real-time operating system as a potential platform for a new generation of products. Technical and legal questions immediately come to mind. For example: Can the software be as reliable as the company will require? If Medtronic decides to use an Open Source system as the firmware in their devices, how do they know they have clear title to it? Is there some way the liability can be removed as a concern? In light of the difficulties within the company of getting sectors to talk with one another, he also wondered if it is feasible to presume that Medtronic's IT people will collaborate with counterparts outside the organization in the Open Source community.

Referring back to Ken Goodpaster's presentation of the four logics, Malcolm wondered if, as an Open Source consumer, Medtronic would have a moral duty to participate in the Open Source community. In other words, he wondered, if non-participation was simply a cause for nagging guilt, or an abdication of responsibility.

With his final question he inherently challenged an assumption that the collaboration associated with Open Source is all done for the side of good. He asked whether or not it can be done in ways that are socially harmful.

**Tony Stanco is a founding director of both The Center for Open Source & Government, and Cyber Security Policy and Research Institute at George Washington University.**

**[Technical difficulties affected the audio record of Tony's presentation. Please refer to the slides for content at this time. Some of Tony's key points are reflected below.]**

Open Source is gaining respect as a technical model for developing software. From both a social and industrial point of view, Open Source is precedent setting because no corporate or governmental structures are instigating the activities. They jumped into the game after the momentum began.

To varying degrees, governments are getting serious about Open Source. The government has some particular obligations to its constituencies; software for government use is special.

Software in a digital society is more than “who can do it more efficiently,” or “who can make the most money.” Capitalist interests are not necessarily ethical in this environment. No single group should have control over software—there are potentially devastating social consequences.

**[Technical difficulties affected the audio record of the Social and Ethical Panel Q & A. Please refer to some key points below at this time.]**

Andras Szakal pointed to a dichotomy of ethics: On the one hand, people tout the necessity for a level playing field, but on the other, some don't want to play in the same game.

Why feel that whatever applies to the rest of the customers shouldn't apply to government in terms of common criteria? (Common criteria aren't about functionality, but rather about the security capabilities of the product—not whether you see the code, but whether it works the way it's supposed to.)

Tony said not to put form over substance. It used to be the job of NSA to review the code, but with Open Source, anyone can review it.

Bruce Perens noted that, in writing the first draft of the Open Source definition, he was exposed to a “moral and ethical” situation that affected his thinking about licenses. His neighbors attached a caveat on a circuit simulation program called Berkeley Spice. The program was never to be used by South Africa, which at the time, had an apartheid regime. The terms of the license went on after the apartheid regime fell, and a Black government assumed power. The attempt to be moral essentially backfired. Bruce concluded that he never wanted to be part of software that was pro-choice or anti-misogynist, or anything else that was flavored by ethics, morality, politics, or religion. He decided that the moral decisions belong elsewhere. He wouldn't, therefore, want to

put caveats in licenses that “you can’t use it if you use it to build bombs” in creating something that may be used by the Federal government.

Amy Marasco added that, if people look at all the things that in this society that are good, they inherently conflict.

### ***The Legal Panel***

*Moderator, Steve Nunn, Vice President and COO of The Open Group*

*Panelists:*

*Lawrence E. (Larry) Rosen, Rosenlaw.com*

*Henry W. (Hank) Jones, III, Intersect Technology Consulting and Law Office of Henry W. Jones, III*

*Tony Stanco, The Center for Open Source & eGovernment, and Cyber Security Policy and Research Institute, The George Washington University*

*Amy Marasco, American National Standards Institute (ANSI)*

Steve opened with facetious questions anonymously submitted to The Open Group web site: If so many technical folks can tell us about licenses, why do we need lawyers at all? And if we don’t need them, are there any new ways of disposing of them? If the answer is that we do need lawyers after all, then are there any new ways of disposing of technical people?

Referring to a meeting in which he participated last week, Steve noted that one of the tracks was “Open Source: Why technologists love it and lawyers are wary of it.” The complexity of Open Source from a lawyer’s point of view centers on the many licenses associated with it. As of the week of June 16, 2003 he noted that there were 43 approved Open Source licenses, and the number is rising. This fact set the stage for the primary questions to be explored by panelists:

- In legal terms, what does, and what does not, fall into the category of Open Source software (for example, shareware)?
- What kind of legal roadmaps are out there? E.g. court cases, arbitrations, and so on
- Who decides whether an Open Source product has been handled properly? As a corollary, how do Open Source advocates know that someone is breaking the rules?
- Are there any key cases that have helped define Open Source?
- Which Open Source software license ambiguities will trigger OSS license interpretation litigation and arbitrations?

**Larry Rosen of Rosenlaw.com**, who is a recognized expert in Open Source licenses and has recently authored two new licenses to address shortcomings in those that exist, launched right into key questions about Open Source software and licenses. In contrast to those who allege there are too many Open Source licenses, Larry argues that there aren’t enough. His rationale for that statement is that the existing licenses aren’t good enough yet—not well written, not clear, not precise.

From the consumer's point of view the bottom line on Open Source licenses is this: They all guarantee anyone, anywhere, for any purpose whatsoever, the right to use the software, copy it, modify it, and distribute those modifications free, or for a fee and the right to have the source code that makes those things possible. Larry's recommendation, therefore, is that consumers have nothing at all to worry about when it comes to licenses. It is only those who intend to distribute software who have to exercise due diligence regarding the terms of the license.

Referring back to the Social and Ethical Panel and discussion of rights-based logic, Larry reminded everyone of the uniqueness and importance of mentioning the Bill of Rights in an Open Source conference. He emphasized that the critical point for Open Source is the desire to have the freedoms associated with that software; licenses are supposed to protect those freedoms legally.

Larry then moved to defining open standards. The best definition of the goal of open standards, he felt, came from Scott Peterson, a lawyer with HP who is widely known for his expertise in Open Source. Scott's belief is that the key is to cooperate on standards and compete on implementation.

Larry reminded the audience of an old saying about setting up tollbooths on the information superhighway. In the Open Source community, that concept of pay-to-use doesn't work and standards can, in fact, be tollbooths. He cited the operation of JBoss Group, of which he is General Counsel, as an example of a company trying to maintain the toll-free spirit of Open Source while also trying to make money. JBoss gives the software away while charging for services, documentation, and training. They gave away two million copies in 2002 and project they will give away double that amount in 2003. He posed the question: Suppose a standards organization charged JBoss one penny for every copy as a "toll" for practicing a j2e standard, for example? It would devastate their revenue stream.

Without denying that revenue generation is necessary to cover development costs, including such things as test kits, Larry maintained that the tollbooth approach was a very big problem for Open Source.

Larry concluded by saying that RAND can't be sharply defined. "Reasonable" is a subjective description. He also challenged how clearly people understand and handle the concept of "non-discriminatory." Nevertheless, he did admit that people with intellectual property protected by patent or copyright had a right to make money from it. The questions are, in the face of a paradigmatic shift prompted by Open Source, "how much money should be made" and "by charging whom." Standards organizations simply must change in light of the shift.

**Hank Jones, both a business consultant and an attorney**, noted that his corporate experience has made him both a defendant and a plaintiff in different circumstances. He first approached the open standards topic by asking the audience how many people had actively participated in standards-setting work? As a corollary, he wondered how many

had to talk to their patent attorneys before and after those meetings. Another question asked how many people had been involved with an Open Source project team.

Hank described standards as a form of gamesmanship. He considers it an open secret that many companies use standards as both a defensive and an offensive weapon. It's well known that companies have filed patents before, during, and after standards-setting meetings.

In recounting some of his own, early experiences in distributing Open Source products, he zeroed in on the point that Open Source triggers problems that are process-oriented and multi-disciplinary. He agreed with Larry Rosen that people are headed for failure if they approach it like religion—as in, there are two religions: mine and everyone else's.

In comparing software to real estate, he emphasized how important it is to know what's owned. With real estate, the protection is a thorough title search to determine if there are easements with pipes or sewers, for example. People should invest the same kind of wisdom and skills in getting to know the property they call software. It is incumbent on the buyer to figure out what the opportunities and obligations are.

The key legal issues, then, come down to the following:

- As a design principal, it's a matter of WYSIWYG versus WYDSIWGY (what you don't see is what gets you); this is the fear of every IP manager
- Sound construction and engineering is a long-term project and design philosophy; software is still young and foolish compared to other industries and developers can learn a lot from the older industries involved with physical construction. Contractors only get paid if they deliver what they are contracted to deliver.
- Plan now for future constituents and unexpected standards-setters in software management; it isn't just computer scientists and lawyers who get to "vote." Hank noted that Open Source teams which he is now working with insurers, CFOs, technical support people, product managers, and more.
- Silos are sickly; multi-disciplinary processes and teams are key to Open Source software; geeks talking to geeks is not a productive model for going forth in Open Source. It's important that a range of people—CEOs, investor relations, and others all need to know what the issues are with Open Source.

He recommended that people in the audience commit to being educators in their arenas. Explain what the noise about Open Source is to colleagues, why people should care, where the software is coming from, how it potentially affects the health of the organization, and so on. He concluded by mentioning a few tools that will make the journey easier, including metrics of litigation costs and overhauled licensing practices.

**Tony Stanco**, also a presenter on the Social & Ethical Panel, put on his lawyer hat as he took a detour from the route that others on the legal panel had taken. He dug into policy issues behind the legal issues. "The policy issues are what drive things. It tells us the reasons why these things are being done."

Tony said it's not particularly useful to look backwards for guidance on Open Source because it's so different from approaches that preceded it. Very simply, it's an unorthodox software development model that few people expected was going to work. Despite that oddness, major IT vendors now have Open Source strategies and Open Source software is gaining market share, so Open Source must be fulfilling some basic economic need. In fact, Tony noted that Wall Street actually "punishes" IT vendors that don't have an Open Source strategy.

For all the wealth that the proprietary software industry has created, it has also created problems, such as monopolies and interoperability problems. The problem might be the proper balance doesn't exist between the rights of the producers and the rights of the users. It is a system problem that must be addressed regardless of whether the approach is proprietary or Open Source.

Turning to the part of the system that addresses intellectual property rights, Tony noted that, because relationships are not based on status but rather on contract, if someone has rights, that person may give them up. One may not, however, get more rights than are allowed by the contract. He says that it is arguable therefore, that Open Source is just a reaction to IP laws that favor software producers too excessively for a correct economic solution; the market is seeking a better solution. In short, 95 years may be an appropriate length of time to protect a book, but that doesn't apply to software. A GPL license, which essentially moves the timeline closer to zero, is the other extreme. According to Tony, Open Source proves that 95 years is wrong, but it doesn't prove that something more closely tied to software product cycles of three to ten years is not better than zero.

He concluded by agreeing with Richard Stallman about the motive for Open Source. The approach is more about freedom, liberty and quality than how to produce better code.

**Amy Maracso, general counsel for ANSI**, offered to shed some light on the murky world of IP—from the perspective of the *de jure*, or formal, standards setting world. ANSI is a private, non-profit organization that administers and coordinates the U.S. voluntary standardization and conformity assessment system. Its mission is to enhance both the global competitiveness of US business and the US quality of life by promoting and facilitating voluntary consensus standards and conformity assessment systems, and safeguarding their integrity.

To start, Amy presented an overview of what the 200+ ANSI-accredited organizations must demonstrate in terms of process and policy to be part of the formal system. The requirements are laden with due process considerations, especially mandatory openness and consensus. All of the different stakeholder interests must be represented on the consensus body. There is also an appeals process and public review. When the standard developed by the organization goes to ANSI for acceptance as an American National Standard, ANSI doesn't review it technically, but simply looks at whether or not the agreed-upon process was adhered to.

ANSI's open standards process has weathered decades of scrutiny and use, so Amy suggested that when many people refer to open standards, they really mean products of the ANSI process, or something close to it. Fundamental elements are that anyone can comment on developing standards and all stakeholder groups must be represented.

Amy then referred to her previous remarks in which she differentiated between open process, as just described, open IPR policy, and openness from a technical point of view.

Amy focused on ANSI's patent policy, which is similar to that at ISO/IEC and the ITU. As a matter of course, patent holders of technology considered essential for implementation of a standard must provide a patent statement telling the community the terms under which they may use it. Some ANSI-accredited groups simply won't accept IP in the standard; others will accept it as long as it's provided with no strings attached, that is, no compensation required.

Acknowledging that Larry Rosen had injected a sharp criticism of RAND, Amy looked to the sense of it. The ANSI policy is that it has to be either royalty-free or RAND, she noted. The rationale behind that policy is grounded in who is participating in standards development. They are technical people and standards development organizations that are very often non-profits. They are not the kind of people who are going to make determinations about the validity of patent claims and how much the technology is worth. The policy, therefore, is set up to establish a "third-party beneficiary relationship," so that those wishing to implement the standard can say to the IP holder, "You've made a representation that you will license royalty-free or under reasonable and non-discriminatory terms." They then have an avenue to pursue that outside of the standards setting process. Otherwise, the technical people who actually develop the standards would be burdened with this and they might feel obligation to bring a legal team to each technical committee meeting.

The Open Source issues are about to collide with ANSI's world, according to Amy, because of ISO/IEC's interest in standardizing Linux.

ANSI has always taken the position that one-size-fits-all rules are not appropriate. Just as there are different standards-setting procedures, there are different IPR policies and it really has to be what's best for that group. ANSI is also initiating, and is responsive to, formal collaborative arrangements with consortia.

## Q&A - Legal

### *Risk Management*

Hank Jones addressed the issue of risk management by first talking about the intimate exposure a customer can get to the code. He then challenged the perception that Open

Source is more risky in terms of warranties and indemnification, noting that a lot of savvy vendors and customers have developed granular, specific questions and representation.

The point was made that, particularly since the September 11 attacks, customers demand assurances about the integrity of their software and their ability to replace it. They want to know, and have assurances in writing, that the software will do exactly what the vendor's marketing people have told them it will do.

Bill Estrem revisited his point about the tragedy of the commons and anti-commons. He indicated that, as Open Source moves toward standardization and different versions become available as a package, some versions of software may contain piece and parts of things that reduce the functionality of the software for others. He wondered if there is a way to have Open Source projects rolling forward into the future where part of the documentation can include some specific traceability of any part that exists.

Larry Rosen reiterated Bruce Perens' point—made often—that Open Source projects are open and he reminded everyone that they were free, for example, to look at the way the Apache project conducts its business. It is easy to know exactly what is in their server, therefore, relatively easy to determine whether or not you trust the product. He joked that one good reason to trust a product from a company that is not open is that it may have billions more in the bank; it would be more profitable to sue them if the product is flawed. In short, he concluded, there are three ways to mitigate risk:

1. Go to the wealthy company to buy software
2. Go to a company that does its business out in the open
3. Buy insurance

Graham Bird referred to historical precedent that simply held a vendor to delivering what is specified in a contract and in product manuals.

Another participant used a building construction analogy to reinforce that point; a demanding customer won't tolerate flawed materials, faulty construction, or a result that doesn't match specified expectations. He focused on the direction he felt the software industry should move, that is, toward guarantees that the end result meets specifications in a contract. Should we go back to the universities and teach people how to build code to precise specs in "an engineer's way," he asked.

You simply can't do that, another countered. It would take an infinite amount of time to test a piece of software to ensure that result.

Larry reasoned that virtually everything we buy has some deficiencies in it—cars, hardware. He emphasized that the focus must be on Open Source process to minimize risks; the assumption is not that processes will be achieved that will eliminate them.

John Terpstra asserted that the English world, including Australia, is very different from the US world: Common law rules and precedent is everything. He cited an odd case in which a company that had been cited for its inclusion of rat meat in product sued a group



distributing food to the needy because they were not a legal entity and therefore could not provide a warranty regarding the quality of the food. He wondered if that set a precedent such that it made the Open Source community vulnerable for distributing free software.

Advancing that question, Chris Hertel of the Samba project asked if he were personally vulnerable in any way by contributing code on a regular basis. He works from home, buys his own equipment, is not subsidized in any way, and has no testing facility.

Chris found out quickly that he actually does run a legal risk. Technically speaking, he is in the software business; Larry clarified that being in the software business has nothing to do with money. He is part of commerce, acting in a business way, and considered a sole proprietorship.

Bruce Perens jumped in with a key fact: He has formed a 501(c)3 (not-for-profit charitable or educational) corporation called Software in the Public Interest ([www.spi-inc.org](http://www.spi-inc.org)) to help address the problem. (Note: SPI is a non-profit organization, which was founded to help organizations develop and distribute open hardware and software. It encourages programmers to use any license that allows for the free modification, redistribution and use of software, and hardware developers to distribute documentation that will allow device drivers to be written for their product.)

Larry Rosen brought the discussion to a close by noting that it is true that individuals may be able to create their own Open Source projects and offer software, but they don't want to offer a warranty on it. On the other hand, legitimate, commercial Open Source projects—Linux, Apache, JBoss and other big projects that do quality software—do not just take software that's tossed over the transom by individual developers. They go through rigorous test procedures. If something is introduced into the product that doesn't work, they fix it. Based on this premise, the example Chris gave doesn't fit the real world of Open Source. He urged Chris to develop a more rigorous quality control process for his work on Samba.

### *Digital Millennium Copyright Act (DMCA)*

There was a question from the floor about the degree to which DMCA is a problem for Open Source. (Note: The DMCA is a US law intended, among other things, to help to prevent people from breaking copy-protection or digital rights management schemes. It effectively restricts the ability of US-based companies to reverse engineer software. Reverse engineering is often necessary if Open Source products need to interact with proprietary software or file formats, since the proprietary software often does not necessarily document how it works.)

Larry Rosen quickly responded that it is, in fact, a very big problem, and that the intent of DMCA is not consonant with that of the original Copyright Act. He urged people in the room to contact their legislators and ask them to rescind DMCA.

Chris Hertel noted that one direct effect is that some Open Source people will not come to the US for conferences because of the ramifications of DMCA. They are specifically concerned about other people being victimized like the Russian programmer detained for five months by the FBI after his presentation at a conference that reflected his use of reverse engineering.

Bruce Perens told the group that SPI provides *pro bono* legal services to developers in the Open Source community and that, in fact, Larry was one of the attorneys who provides them. He recommended using this service in the early stages of a project, not as an afterthought if and when there is a problem.

He also threw in an interesting thought regarding the source of some of the code: Programmers who are legal minors. Issues related to their contributions are not just legal, but in some cases, outcroppings of a lack of maturity regarding content. He emphasized the need to educate the community at large about the range of issues.

### *Open Source, Open Standards and IPR*

Steve Nunn set the next topic as the relationship between Open Source and open standards: “Is the formal standards world threatened by Open Source, and if so, what do we need to do about it?” He directed it first at Amy Marasco as the representative from ANSI, the lead organization supporting the formal standards process in the US.

Amy responded that ANSI does not support just one system for producing standards—a one-size-fits-all—but rather matching the need to the process. She noted that ANSI is exploring ways to work with consortia and that Open Source is making its way into the formal system through JTC 1 and its project to address Linux standards. Amy also agreed with Bruce that educating the interested parties is a big part of the solution of understanding common objectives.

Larry took a different angle by saying that they are scared and should be scared. The experience of W3C is an important one. He was referring to the W3C adoption of a patent policy that included RAND. The Open Source deluged W3C with e-mails of protest, saying that it was not acceptable to “proprietaryize” the web, which had been created in an Open Source way. He reiterated his assertion that the paradigm is changing.

Amy maintained that, just because the paradigm is changing, that does not mean that the formal standards community does, or should, feel threatened. ANSI has changed its patent policy in the past, and it could change it again in light of evolving paradigms. She then differentiated between standards setting bodies, and how they handle intellectual property rights, and individual companies within those standards bodies, and what their preference is for establishing IPR rules.

### *Wrap-Up Session*

Allen invited the moderators of all four panels for their conclusions, in their respective areas, on what the primary action should be coming out of this conference.

### *Business Panel*

Graham registered some surprise that the conference revealed, “We aren’t as far down the track with Open Source” as he had thought. He emphasized the serious interest in, and need for, probing issues further so that people had a practical understanding of the most productive ways to integrate Open Source into their businesses.

Chris Hertel interjected that he felt that a disconnect was evident between the developer community and commercial interests. As a programmer, he said he just wants to write code; he asserted that he does not want the onus to make the output “a real product” on people like him. His response is, “You make it a real product.” He stressed that developers don’t necessarily have the same goals in mind as those of you with commercial interests.

Expressing appreciation of that perspective, Graham pointed to the stated desire of many developers to make a contribution that “makes a difference.” He suggested that making that difference may involve listening to what the marketplace is saying its needs. In that sense, strategic marketing has an important role in bridging the gap between objectives in Open Source.

### *Technical Panel*

Terry logged the number one issue as defining an open standard and identifying the process, or processes, to produce it—how to make sure that requirements make it into standards and making it clear who should participate in the process. He called for the creation of a standards business scenario.

### *Social and Ethical Implications Panel*

Bill flagged the question “Can we be innovative in a six sigma world” as a key one emerging from his panel. He felt that it was incumbent on industry to look at the impact on innovation of an obsession with perfection of process. He also noted that the issue of personal accountability was key in Open Source and that developers needed to explore that as it pertains to producing quality output.

Russ Richards of DoD added that he felt it would be a useful exercise to take the four “logics” and structure a session around them as they related to various aspects of Open Source.

Allen Brown added that different concepts related to the Open Source discussion, such as privacy, play out differently depending on the country and culture. Discussions of social and ethical implementations of Open Source would therefore produce different output depending on the venue in which they occurred.

He also asked the audience whether or not The Open Group's posting on the web of the findings of such a discussion would stimulate feedback. Bruce suggested that it might also make a provocative article.

### *Legal Panel*

Steve concluded that warranty and indemnification stimulated the most animated discussions and felt that both sides made eye-opening observations (that is, the developers and commercial interests). He tied it in with the principal question in the business panel and suggested that effectively addressing the warranty/indemnification issue is central to making Open Source "ready for prime time" in the enterprise.

He added that the number of Open Source licenses also embodied a mandate for the legal community—it is important to help users of Open Source understand "where to start" in terms of licenses. He felt that The Open Group might initiate an activity to record best practices in this area, and coordinate its efforts with the OSI.

Bruce expressed support such an initiative.

### *Closing Remarks*

Allen asked everyone to recall the frescoes of the seven virtues that are in the entrance hall of the building adjacent to the meeting room—justice, charity, hope, courage, faith, prudence, and temperance. He asked if the consensus was that Open Source might capture all of them. The conclusion: All but temperance. "But we have added one," he said. "Sharing."

**Appendix A**  
**Table of Contents and Executive Summary of the Eurim Report**  
***Draft Status Report on Open Source Software***  
  
From the Open Source Subgroup of  
EURIM's Modernising Government Working Party

**CONTENTS**

- I EXECUTIVE SUMMARY**
- Abstract
  - Summary Points
- II INTRODUCTION**
- What is Open Source Software?
  - What kind of applications is OSS used for?
  - *Licensing Issues*
  - *Schools of thought and ideologies*
  - How the different models operate in the market place
  - Why has Open Source developed such a popular following?
- III WHAT IS BEING DONE BY GOVERNMENT?**
- Current situation – what is the policy?
  - *How is policy being implemented?*
    - i. Implementation of OSS policy in the UK
    - ii. Implementation of OSS policy in the EC
- IV THE ISSUES**
- Confusion
    - i. Over basic terminology
    - ii. Over licensing regimes
    - iii. Over polarised opinions
    - iv. Over advantages and disadvantages
  - Arguments for and against OSS
  - Evaluating the business case
  - Making decisions about OSS
- V CONCLUSION**

## ***EXECUTIVE SUMMARY***

### ***Abstract***

This paper sets out to explain the background to Open Source Software (OSS) – what it is, what it is not, to summarise some recent developments in both the policy arena and the marketplace, and to provide some basic facts about the products available. Contributors include IBM, Microsoft, OeE, BSI, Fujitsu Services.

### ***Summary Points***

OSS is a significant and serious competitor with commercial solutions in many ICT applications and is now taking a significant market share in some parts of the software infrastructure market.

OSS is a tool and a methodology, one of a number of software development models now available. OSS must be viewed and handled as such, not as a cure-all.

The debate on the respective merits and de-merits of the commercial and open source software models has become ideological and polarised to an unhealthy degree. However, there are signs that pragmatic approaches are increasingly being adopted by the different stakeholders.

There is room for both OSS and commercial models of software development to co-exist. Current Government procurement policy states that there should be a level playing field for procurement so that both can indeed co-exist at least in terms of Government buying.

The market provides a substantial middle ground where vendors, whether traditionally OSS or commercial\* are offering combinations of OSS and commercial packages, are including OSS elements in commercial software or are packaging OSS products commercially.

As the UK moves increasingly towards a knowledge-based economy, so intellectual property issues become more important.

Parliamentarians should take an interest in these issues, as with many other developments in IT and technology they have important implications for economic and societal development.

Parliamentarians must be aware that the situation is complex, and there are too many considerations, for one model to be generically favoured over another. Public sector procurement decisions should be made on a value-for money (VFM) basis.