

A history of IBM's open-source involvement and strategy



P. G. Capek
S. P. Frank
S. Gerdt
D. Shields

IBM has embraced both the concept and the reality of open-source software (OSS). While this may seem surprising for a corporation with a large traditional software business, a path has been pursued allowing the maintenance—indeed, the enhancement—of existing business while deriving benefits from OSS, making significant contributions to several OSS projects, and initiating new ones. We describe some of the considerations which led to this approach, as well as some of the issues that impinge on its execution.

The origins and principles of free software and of open-source¹ software (OSS) may lead the casual observer to conclude that they are a world apart from—if not opposed to—more traditional software development, use, and evolution. An alternative view sees OSS as essentially an alternative business model which provides types of flexibility, opportunity, and benefits different than those provided by the conventional model. IBM was among the earliest of the major computer companies to embrace open-source software and was probably the first to realize that doing so could be consistent with our business goals. Indeed, a problem with which IBM has long contended is that of how to provide to our customers internally developed software that was not planned to be a product, without the inevitable support and product issues.

IBM's business strategy has long been centered around open standards, both for hardware and software, wherever that is feasible and practical, and IBM has taken an active role in the development of standards related to the company's business.

Early on, it was perceived that OSS offered interesting opportunities with respect to these activities, and that as a result, there were many reasons to investigate how best to integrate open-source software with our business.

BEGINNINGS

In December of 1998, an effort was first made to understand the broad strategic implications for IBM of open-source software. At that point, it was clear that the OSS phenomenon was taking hold in a substantial way. Most visibly, Linux** was starting to appear widely in the media, but more importantly, parts of our customer organizations were starting to pay attention, with Linux reportedly being used in some cases without the involvement or blessing of corporate IT organizations. Quickly,

©Copyright 2005 by International Business Machines Corporation. Copying in printed form for private use is permitted without payment of royalty provided that (1) each reproduction is done without alteration and (2) the Journal reference and IBM copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free without further permission by computer-based and other information-service systems. Permission to republish any other portion of the paper must be obtained from the Editor. 0018-8670/05/\$5.00 © 2005 IBM

we realized that whether this evolved into an important force or whether it remained a minor fad,

■ IBM's business strategy has long been centered around open standards ■

the potential was such that it was important to understand its implications for our customers and for us and be able to respond appropriately. Before 1999, our involvement was on a case-by-case basis.

It was apparent that some new IBM employees were aware of OSS and the collaborative aspects of the OSS model. Students newly completing their Ph.D. degrees joined IBM after having done their thesis work and making it available to the world, often under an established open-source license. Allowing them to continue working in their thesis area and continue to contribute to the open-source project that they had initiated while in school stretched the limits of IBM's standard business practices. At the same time, it provided an early warning of the issues involved with OSS.

An important issue was the quality of software that was produced by open-source communities and their collaboration. Much of IBM's product software development was historically quite structured, with substantial initial planning and design, followed by implementation, unit and system testing phases, and of course ongoing support and maintenance. Many at IBM had the impression—partly from what appeared in the business and technical press—that open-source software efforts were closer to the other end of the spectrum in terms of structure and management discipline, and they were accordingly skeptical that the quality of the open-source software produced could be sufficient to be relevant to us and our customers.

These early fears turned out to be unfounded. Even at that time (ca. 1999), the quality of the software from the open-source projects investigated was impressive. It was clear that this development style attracted very skilled developers, and that the overlap between developers and users of a particular OSS project made possible excellent and open communication, rapid development cycles, and intensive real-environment testing, ultimately pro-

ducing software that was often very good and sometimes excellent by our standards. At the same time, it was immediately clear that there were important areas where IBM's large and excellent technical community could make significant contributions, having substantial experience, and in doing so, our customers could be helped to reap the benefits of our expertise in an open context. In more recent years, the possibility of inverting the model has been investigated, whereby our proprietary development activities can benefit from what has been learned from the open community.

As this work was going on, its urgency was growing. IBM had released a binary-only version for UNIX** of Jikes, a compiler for Java**, through our alpha-Works* Web site² in early 1997. This was quite successful, but its success was only a prelude to the release of a Linux version of the same code in mid-July 1998, which was downloaded at seven times the rate of the non-Linux versions. Requests for the source code followed rapidly, and at roughly the same time as we were beginning to understand the larger, strategic implications, the source code for Jikes was released in late 1998 under a rather liberal license based on the Apache** license.³

One of our first and most memorable experiences with OSS followed the Jikes source code release. Within eight hours of the release, a programmer in California sent an e-mail to the Jikes authors containing a non-trivial enhancement to the compiler, one which required investing some time and effort to understand the code.

From the outset, it was clear that a host of legal and business considerations needed to be understood if IBM was going to participate in any OSS activities in a meaningful way. Much of the participation and development of OSS at that time was done by individuals acting on their own.⁴ There were some early efforts that were more organized and which involved small companies, but these were, for the most part, companies organized around their open-source participation. A few notable examples included companies that were using open source in their own operations and contributing enhancements and development to it for the broader good.

IBM, of course, had a large software business, which could not be put at risk; therefore, it was important that any risks associated with OSS be identified, and

the legal, strategic, and business issues surrounding open source and its licensing be understood. Where needed, procedures would have to be established to ensure that our participation was principled and appropriate.

We were fortunate that we could draw upon a number of resources for our education, both in print^{5,6} and by interviewing knowledgeable people in the field. From the beginning, the participation of attorneys, as well as business and technical people, was encouraged.

Ultimately, we had to judge as well as we could how the industry would evolve and determine how to respond in terms of our OSS activities. We judged, even as early as 1999, that Linux had at least a modest chance for significant success and that our involvement could significantly improve that chance. We also saw in Linux the possibility of having a unified operating system on our platforms in a way that seemed achievable (and has since been largely realized). Our emphasis with respect to Linux has been to make it fit better into the enterprise environment.

More generally, a strategy was planned that allowed us to add value for our customers in the areas where our ability to do so was greatest. This was clearly in the broad area of what is called middleware, and not in operating systems, because our enterprise customers benefit more directly from middleware functions than from operating-system functions; analogous statements can be made in other areas. Consequently, our strategy for open-source participation was one which effectively minimized the distinctions at the operating-system level and allowed us to retain the ability to differentiate where we could have the greatest impact.

THE KEY LEGAL ISSUES

Because software is a form of tangible expression, it is legally covered by copyright law. As with any such work, the copyright owner, initially the author, has the exclusive right to copy and distribute his software work and to prepare derivative works thereof. A copyright owner may choose to grant all or some of these rights to others by means of a license. A software license may also include grants of other intellectual property rights, such as patents or trademarks, and it may include constraints or requirements that apply to a recipient. Open-source

licenses, of which there are many, are simply copyright licenses that generally provide for the right to freely modify and redistribute the software and that may contain other provisions, such as a patent license grant.

Complexity sets in with software because most substantial open-source software has many authors and is developed in a collaborative and informal manner by people with no particular legal relation

■ An important issue was the quality of software that was produced by open-source communities and their collaboration ■

ship. For these projects, it is often difficult years later to know reliably whether the person granting a license had the right to do so. For instance, was a particular contributor the author of the code, and did he have the right to grant a license, or did his employer acquire that right when he wrote it? Although some projects, including those under the Free Software Foundation, have long required assignment of copyright by each contributor including written signatures, this has not been a universal practice. Recently, more software community leaders have recognized the importance of creating clarity of code “pedigree” and rights, and IBM has worked to assist some open-source projects to increase the rigor of their processes in this area. Examples of these efforts are the Linux kernel and its Developer’s Certificate of Origin⁷ and the Apache Software Foundation and its Contributor License Agreement.^{8,9}

Another legal consideration was the proliferation of licenses used for open-source projects. None of these licenses had been interpreted by any court, and they varied greatly in terms of their legal robustness and completeness. Many of them were unclear with respect to the granting of intellectual property rights. As a commercial organization, we felt it was important to encourage a model in which commercial products could be based on open-source efforts, and we needed to identify a license that would permit such a model. Thus, IBM created, used, and encouraged the use of, what is now

known as the CPL, or Common Public License. This license has been well received by the community, and its use seems to be increasing.¹⁰ It has been certified as an open-source license by the Open Source Initiative. Our goals in creating this license were to provide a means for commercial organizations to base products on open-source efforts, to encourage a common OSS practice of making modifications and enhancements available as source code, and to provide a model which could help to shape other open-source licenses. In our opinion, this license provides a good balance between open-source and commercial efforts and encourages enhancements to open-source projects.

Patents are an important aspect of software today and must be considered when OSS is developed. When OSS is created and licensed, it must, as a practical matter, carry with it a grant of license to any patents concerning the software that the author holds. Doing otherwise creates an untenable situation, wherein any users of that OSS may become inadvertent infringers of the patent. Some licenses (CPL is an example) include an explicit grant of a patent license, but most do not.

IBM has recently shown its strong support for OSS by granting a license, to any open source effort, for 500 IBM patents.¹¹ It is hoped that other patent holders will join IBM in establishing a “patent commons” for the benefit of OSS and to encourage innovation. IBM continues to strike a balance between proprietary and open-source efforts, retaining some patents for use by itself and its licensees, and avoiding the use of these ideas in its open-source efforts.

Lastly, a variety of considerations arise in establishing and maintaining good legal practice between proprietary and open-source activities. These considerations can be dealt with through education of employees, through proper procedures and documentation, and by participating in OSS efforts through protocols that provide sufficient insulation and isolation from a company’s proprietary efforts, while allowing them and the community the benefits of OSS involvement.

BUSINESS CONSIDERATIONS

In exploring the world of open-source software, we worked quickly to understand the legal issues, both because IBM attorneys participated and cooperated

with us fully and because most of the issues were fairly straightforward, being based on copyright law, licenses which we could read and evaluate, and contract law. The business considerations were not nearly as clear cut; they involved considerably more analysis and guessing at likely directions in which the industry, and open-source efforts, might move. Even then, it was difficult to be completely confident in our conclusions.

What would an extremely successful open-source movement mean to IBM’s software business? Even at the time that we were examining these issues, we could see the first hints of the increasing importance of service and support in our key markets. While that trend was itself unlikely to eliminate licensing of software and the resulting revenue as a business model, OSS clearly could, in the medium to long term, have an effect on the balance. A collection of successful open-source projects could clearly benefit IBM’s customers under the right set of circumstances. IBM, under that assumption, would be wise to participate. But, of course, further complicating the discussion was the fact that IBM’s own actions could have a significant effect on the success of various projects, and so it was important to try to understand not only what we would like to happen, but also what was likely to happen.

In early 1999 when we were exploring these issues, open source was successful in a set of important but somewhat circumscribed areas that can be categorized as “infrastructure.” This category includes extremely important, but often hidden, software, including, Linux (operating system), Apache (Web server), DNS (domain name service, a key part of the software which “runs” the Internet), GCC (the GNU compiler collection), and others.

Software that is directly visible to end users, or even to application developers, such as the middleware and application software which is of most concern to IBM’s key market and decision makers, was not a major area of open-source activity, although this is beginning to change. As a result, open source did not pose an immediate threat to our existing businesses, and in fact, our products could benefit from supporting and building on open source. The key example of this is the WebSphere* Application Server, which is built using the Apache HTTP (HyperText Transfer Protocol) server as a key component.

The other aspect which was important for us to understand was how we might be able to use participation in OSS activities to the benefit of our customers and ourselves. Clearly, open source provided an avenue by which IBM could release software with much less overhead and expense than is normal for anything which is a product. The alphaWorks Web site was initiated in August, 1996 as a step in this direction, but it had different goals and was limited in that only binary executable programs were distributed in this way, and licensing was normally limited to 90 days. Open source provided a much broader avenue to reach the community. Given IBM's stated strategy of encouraging and using open standards, it also seemed clear that open source provided a way in which we could release implementations of proposed standards for the community to test and explore, and thus lead development efforts effectively.

We also saw—and have since implemented—open source as an important way to increase our development leverage. There are certain areas of software that are extremely important but are not in and of themselves very profitable. The key example of this for IBM is development tools. Because the usability and effectiveness of application development tools strongly affects the value and viability of products like Websphere, it has always been important, though difficult from a business point of view, to invest in these tools. Open source provides an excellent way to allow the community to contribute, thereby naturally helping these tools evolve in the way the user community desires, while enhancing their value in support of the product which is important to IBM. This can effectively shorten the “need-implement-use” loop, compared with what it might be with the conventional product cycle. The Eclipse open source project, with its affiliated eclipse.org governance body,¹² has provided a very successful model for this kind of effort and is further discussed later.

Finally, the importance of the community to the open-source idea is critical. This is not something which we fully appreciated immediately, but which has become increasingly apparent over time. An open-source community, and especially its leadership, plays the roles of management and team leaders in a business. The community is in some ways more important even than management because the economic issues and motivations are

different, and therefore, important considerations such as loyalty, satisfaction, and continuity play out differently. The open-source effort is held together by shared interest, and it is important for the leadership of an effort to preserve that.

MAKING OPEN SOURCE HAPPEN

In March 1999, a report was prepared summarizing our findings and presented to the Corporate Technology Council, a management group that governs key IBM strategy and business decisions. It was well-received although a number of questions were raised, particularly about business considerations, which involved more homework and analysis. Ultimately, a small group was set up within the IBM Software Group organization to oversee IBM's open-source activities, formalize the goals, create educational materials and provide training, and manage the day-to-day aspects of our activities. This included making sure that appropriate approvals were granted before any IBM team externally participated in an open-source activity and that team members received appropriate education.

The immediate effect of the report and the work that followed was to encourage IBM executives to think about open source as an adjunct to our existing business methodologies, offering a new degree of freedom. In particular, the report clarified how open source could help us in establishing open standards. Of course, assimilating this took time, but our participation accelerated, as demonstrated by the January 2001 announcement at LinuxWorld that IBM planned to invest \$1 billion in Linux over the next three years.

IBM'S OPEN-SOURCE STRATEGIC GOALS

Since their establishment during 1999, IBM's strategic goals for open source have remained consistent. They are:

- To support rapid adoption of open standards by facilitating easy access to high quality open-source implementations of open standards in order to speed industry adoption. A primary goal is to encourage open-source implementation of open standards and thus use open source as a way to support our business and strategic goals.
- To use open source as a business tool by keeping the platform open and taking advantage of new business opportunities. By creating more open

opportunities, we encourage choice and flexibility in responding to customers' needs in typically heterogeneous environments.

- To enhance IBM mind share, creating a preference for IBM brands by associating them with successful OSS projects and building relationships with a broad spectrum of developers. We contribute to key OSS projects that are functionally connected with some of our key products. The joint participation of commercial developers and independent OSS developers creates a synergy that enhances the open-computing "ecosystem."

To summarize these goals, IBM views open source as a tool or technique to be used, where it makes sense to do so, to enhance our business and that of our customers. We strive to do this in a way that makes significant contributions to open-source

■ Our proprietary development activities can benefit from what has been learned from the open community ■

communities and projects and are often able to do that. We acknowledge that we benefit from the open-source efforts of others, but are, on balance, a net contributor.

KEY FOCUS AREAS

The following summarizes our major strategic open-source activities. Several of them are covered elsewhere in this issue of the *IBM Systems Journal* in greater detail.

Linux

Linux is an important corporate initiative that spans our product lines in servers, software, services, and storage. IBM efforts have clearly accelerated Linux acceptance in the enterprise by increasing scalability to 8-way SMPs (symmetric multi processor systems) and beyond and improving reliability through efforts in stress testing, defect management, documentation, and standards. The broad adoption of the Linux Standards Base certification program is a key step in solidifying Linux in the eyes of ISVs (independent software vendors).

Full-time IBM Linux developers from IBM's Linux Technology Center (LTC) have become accepted

and valued contributors in virtually all aspects of the Linux operating environment, including the core kernel itself. A majority of contributions developed and delivered by IBM are being accepted into the base code streams, and those contributions have demonstrably accelerated the maturation of Linux in the enterprise. Examples include improved scalability, networking, serviceability, system performance, I/O performance, usability, availability, standards, security, network security, storage, and file systems.

IBM's vision for the future of Linux as a mission-critical, pervasive component of the IT infrastructure has permeated the Linux industry and is showing up in the Carrier Grade Linux and Data Center Linux roadmaps from the Open Source Development Laboratory—efforts that have made technology adoption by distributors more predictable and consistent. IBM has also worked with Novell/SUSE and Red Hat in getting Linux certified on the Common Criteria, a security classification especially important in the government and public sector.

Apache

The Apache HTTP server and many XML (Extensible Markup Language) projects have become de facto standards, and we are leading numerous projects in the Web services area to help develop this emerging business. The Apache community develops open source implementations of key Internet technologies like the Apache HTTP server (a key infrastructure component of WebSphere), XML tools, Web services components, and the Jakarta set of Java-based Web technology tools. IBM joined the Apache community and became a regular and influential contributor in order to help Apache produce early open-source reference implementations of new Web technologies that would speed the adoption of the associated open standards by putting code into the hands of developers.

IBM enjoys an excellent relationship with the Apache community and currently has two (of nine) members on the Apache Software Foundation board of directors. IBM engineers are valued contributors to key Apache projects, including the Apache HTTP Server. They are also project leaders and maintainers of Xerces (XML parser), Xalan (XML transformer), and SOAP (Simple Object Access Protocol). In the Web services area, IBM

has led the creation of subprojects including Web Services Invocation Framework (WSIF), Web Services Inspection Language (WSIL), Web Services for Remote Portlets (WSRP4J), and Remote Portals (Pluto).

A new OSS project within Apache is Derby, a Java database based on IBM's Cloudscape technology. It was accepted as an incubator project in August 2004. The goal is to enhance the attractiveness of the Java environment by providing a relatively lightweight, zero-administration database system for incorporation into Java Web servers and other applications.

Eclipse

Eclipse 1.0 was released in November 2001 (supporting Linux and Windows), and Eclipse 2.0 was released in June 2002, adding support for AIX*, Solaris**, and HP-UX**. Usage of the Eclipse platform is very popular, with over 20 million download requests as of August 2004. The Eclipse community continues to evolve a rich client platform, instead of only a tools integration platform, as new projects include more tools for developing J2EE** applications and support.

The Eclipse community maintains a number of growing project areas. The main project maintains the Eclipse platform, which includes the starting code base, Java tools, and the plug-in environment. The tools project includes the popular Graphical Editor Framework (GEF) and the subproject Hyades, which is a platform for automated software quality tools. The technology project includes AspectJ (an aspect-oriented language extension to Java) and the Equinox subproject, whose goal is to broaden the range of Eclipse platform runtime configurations. A new project area is the Web tools platform, creating specialized and differentiated offerings for J2EE and Web-centric application development.

The expansion of code projects is evidence that Eclipse is evolving in the life cycle of a successful OSS code development community. As Eclipse adoption grows and gains traction, a de facto industry standard is established, and a growing set of skills is facilitated in the programming community through the schools, nontraditional IBM programmers and customers, the open-source community, and corporate IT.

In February 2004, Eclipse became an independent organization. Founding board members of the not-for-profit corporation include IBM, HP, Intel, Ericsson, MontaVista Software, QNX Software Systems, SAP, and Serena Software. The Eclipse community is growing beyond the original contribution of IBM-created code to attract more subprojects and third-party contributors, a critical element in developing an interactive, supportive, and diverse OSS community.

Globus alliance

The goal of the Globus Alliance Project is to promote grid computing and provide the grid-computing community with an open-source solution for the adoption of grid computing. Initially, each grid-computing implementation provided a grid-computing infrastructure that was based on unique commercial programming models, which created issues for operating across heterogeneous grids.

■ A strategy was planned that allowed us to add value for our customers in the areas where our ability to do so was greatest, namely, in middleware ■

Customer and ISV applications had to be specifically factored to run on each commercial grid implementation. In reaction to this, communities that are implementing grid-computing technology today are calling for standards that will enable applications to share system resources (CPU, memory, storage and networks) regardless of server platform or grid middleware infrastructure.

IBM has assumed an industry leadership role to promote the definition and implementation of standards for grid-computing environments based on the Open Grid Services Architecture (OGSA). IBM is participating in the Global Grid Forum (which is responsible for defining standards for grid computing) and the Globus Alliance Project (recognized as a leader in providing open-source solutions for grid computing). Our goal is that ISVs and customers refactor their applications (i.e., remove or rewrite old code) to run on implementations based on the OGSA.

IBM has made tactical OSS contributions to the Globus Toolkit to support eServer* zSeries*,

pSeries*, iSeries*, and Linux. On the strategic front, IBM has contributed code to Globus Toolkit 3, supporting the OGSA and the OGS (Open Grid Services Infrastructure). In addition, IBM is refactoring our core middleware foundation technology and key autonomic computing technology on the OGSA in order to support IBM's on demand computing initiative.

MANAGEMENT OF IBM OPEN-SOURCE ACTIVITIES

The Open Source Steering Committee (OSSC), an IBM internal board, oversees open-source activities and reviews all planned external uses of open source. The OSSC Program Office provides ongoing education to teams across the corporation utilizing our Open Source Participation Guidelines (OSPG). An internal Web site is maintained with pertinent information on OSS and the OSSC review and approval process. In addition, we sponsor an open-source zone on our developerWorks site (<http://www.ibm.com/developerworks/opensource>) that highlights our various activities in open-source communities and provides additional content to developers.

OSS interest within IBM continues to grow. To manage the growth, there is continued focus on process improvements. For those development teams considering an open-source proposal, there are templates that address questions covering five major areas: business, strategy, law, technology, and community. In addition, there is a set of legal "due diligence" guidelines to ensure that our site attorneys perform a consistent review in evaluating a proposal.

IBM's experienced teams, including Apache, Linux, and Eclipse, have developed competence and in some cases excellence in mastering open-source development from both a technical and a management perspective. We now can leverage that experience in other ways, such as using some of the open-source development processes in our internal processes of developing commercial software products. We are encouraging the usage of IBM's Internal Open Source Bazaar (IIOB) for some development projects, in order to leverage the open-source development methodology. The use of the bazaar can also be an effective mechanism as IBM develops a componentization strategy in both existing and new IBM products.

THE FUTURE OF OSS

We often consider the long-term outlook for IBM's OSS strategy. Will OSS pervade all aspects of software development? What is the outlook for non-OSS software? While it has been said that prediction is very difficult, especially of the future, we can offer some thoughts.

Open source has been most successful with programs whose appeal is broadest and where there is commonality of interest between developers and users. This is most apparent in the success of Linux, Apache, and Eclipse. The market for many of IBM's products is narrower—fewer organizations require the kind of enterprise-level software for which we provide the greatest value. We believe our OSS model is sustainable although we accept that it will need to evolve in its details.

We foresee an ongoing focus on adding value for the customer, particularly at higher levels in the software stack—whether in proprietary or open software—as lower layers become increasingly open and standardized.

CONCLUSION

In summary, we have learned a great deal during the past several years to enable what we believe is a balance between our traditional commercial business and our engagement with the open-source community. At the beginning, we recognized the need to address the strategic implications from both business and legal points of view. In addition to benefiting from the work of the greater community, we saw an opportunity—if not a responsibility—to actively participate and contribute toward the vibrancy of key OSS projects. We will endeavor to maintain this symbiotic relationship to the benefit of our customers, business partners, and shareholders.

*Trademark or registered trademark of International Business Machines Corporation.

**Trademark or registered trademark of Linus Torvalds, The Open Group, The Apache Software Foundation, Hewlett-Packard Company, or Sun Microsystems, Inc.

CITED REFERENCES AND NOTES

1. We are well aware of the distinctions between "free" and "open source" software, but will treat them equally for the purposes of this paper.

2. IBM alphaWorks: Emerging Technologies, <http://www.alphaworks.ibm.com>.
3. Licenses, The Apache Software Foundation, <http://www.apache.org/licenses/>.
4. Although we have no hard facts to offer, it is clear that a very substantial portion of the work done today on the major open-source projects is done by people paid by their employer to do so. IBM, for example, has more than 700 people contributing to such projects, primarily Linux, Apache, Eclipse, and Globus. We believe the often quoted notion that such software is written primarily by people working *gratis* for the general good is false although clearly there are substantial and important contributions which such people make.
5. *Open Sources: Voices from the Open Source Revolution*. C. DiBona, S. Ockman, and M. Stone, Editors, O'Reilly Open Source, O'Reilly & Associates (1999).
6. E. S. Raymond, *The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*, O'Reilly & Associates (1999).
7. "OSDL to Support Enhancements to Linux Kernel Development," *OSDL Web site* (May 24, 2004), http://www.osdl.org/newsroom/press_releases/2004/2004_05_24_beaverton.html.
8. Licenses—Contributor License Agreements, The Apache Software Foundation, <http://www.apache.org/licenses/#clas>.
9. For programs developed within IBM, procedures are in place that require the authors to sign a "Certificate of Originality" stating that the code of the product is their own creation and not copied from other sources.
10. Eclipse started with the CPL and now uses a slightly modified version known as the Eclipse Public License.
11. *IBM Statement of Non-Assertion of Named Patents Against OSS*, IBM Corporation (2005), <http://www.ibm.com/ibm/licensing/patents/pledgedpatents.pdf>.
12. Eclipse.org Main Page, <http://eclipse.org/>.

Accepted for publication November 24, 2004.

Published online April 7, 2005.

Peter G. Capek

IBM Research Division, 1101 Kitchawan Road, Rte 134/P.O. Box 218, Yorktown Heights, NY 10598 (capek@us.ibm.com). Dr. Capek is a research staff member at the IBM Thomas J. Watson Research Center, presently on the staff of the IBM Senior Vice President of Research. His research interests include computer architecture, operating systems, and computer security. He was involved with the early formulation of IBM's open-source strategy. He is a member of the IEEE, ACM, and IBM Academy of Technology.

Steven P. Frank

IBM Software Group, 1551 South Washington Avenue, Piscataway, NJ 08854 (spfrank@us.ibm.com). At the time this paper was written, Mr. Frank was a Program Manager with the Open Source Program Office in the Software Group Technical Strategy organization providing leadership across the corporation in implementation of IBM's open-source strategy. He is currently in the Lotus Division as a member of the IBM Workplace Client Technology team, which has built a rich client platform and applications based on Eclipse. He continues as a Lotus representative on the Open Source

Steering Committee Core Team. Prior to joining IBM in 1999, he spent several years with Lucent Technologies as a systems engineer and in the productization of the Inferno operating system (based on Plan 9, now available as open source), developed in Bell Labs under the leadership of Dennis Ritchie, where Mr. Frank first learned about open-source-style software development.

Steve Gerdt

IBM Software Group, 5600 Cottle Road, San Jose, CA 95193 (gerdt@us.ibm.com). Mr. Gerdt is currently Program Director of IBM's Open Source Program Office, where he manages IBM strategy for open-source initiatives, manages the review of planned usages and participation in open source, and provides leadership for involvement in key open-source communities. He has been a leader of IBM's Open Source Program Office since 2000. Prior to his current role, he held various development, architecture, and management roles at IBM in storage software, delivering enterprise storage systems from the IBM 3990 storage controller and IBM 9340 DASD to the RAMAC® and Total Storage® Enterprise Storage Server disk array systems. He received a B.S. degree in computer and electrical engineering from Purdue University in 1984 and an M.S.E.E. degree from Stanford University in 1989.

David Shields

IBM Software Group, Route 100, Somers, NY 10589 (shieldsd@us.ibm.com). Dr. Shields has a B.S. degree in mathematics from the California Institute of Technology and a Ph.D. degree in computer science from New York University (NYU). He did research in programming language and tools while at NYU, including the SETL project, which developed an influential language based on set theory, and the NYU/Ada project, which produced the first validated Ada compiler. He joined IBM Research in 1987, initially as part of the PTRAN (parallelizing translation) project, well-known for its pioneering work in this area. He and Philippe Charles wrote Jikes, a Java compiler, that evolved into IBM's first open-source project (see <http://www-124.ibm.com/developerworks/projects/jikes>). He led the effort to release the code in source form and ran the Jikes open-source project for its first year. He has also worked on Eclipse, another of IBM's open-source projects. For the last two years he has worked as a member of the interdisciplinary team charged with the management of IBM's open-source activities throughout all divisions and geographies. ■