

OpenEC/B:
**A Case Study in Electronic Commerce and Open Source
Software Development**

Walt Scacchi
Institute for Software Research
and
Graduate School of Management
University of California, Irvine
Irvine, CA 92697-3425 USA
Wscacchi@ics.uci.edu

July 2002

© Walt Scacchi, 2002

This report can be found on the Web at:

<http://www.ics.uci.edu/~wscacchi/Papers/EC/OpenEC-July2002.pdf>

***Acknowledgements:** The research described in this report is supported by grants from the NSF Industry/University Research Cooperative CRITO Consortium, National Science Foundation #IIS-0083075, and from the Defense Acquisition University by contract N487650-27803. No endorsement implied.*

Executive Summary

This study investigates how to understand and transform an organizational system for internal or external operations using Open Source E-Commerce or E-Business capabilities. This entails a case study within one firm that has undertaken an organizational initiative to develop, deploy, use, and support open source software systems for Enterprise Resource Planning (ERP), E-Commerce (EC) or E-Business (EB) services. The objective is to identify and characterize the organizational resources and development capabilities that lie at the center of the initiative. By learning what these resource are, and how the are arrayed into product development capabilities, can give executive decision-makers knowledge needing to determine whether or not to undertake a pilot study or product development effort that exploits the potential of open source EC and EB resources.

The study provides a survey of large, mid-size, small, and start-up companies that are making strategic investments in open source software systems for ERP, EC, and EB products or service offerings. Large companies include IBM, SUN, Hewlett-Packard, Apple, SAP, and AOL within the IT industry, and also Dresdner Kleinwort Wasserstein (DKW) and Barclays Global Investors (BGI) in the international banking industry. Mid-size companies include Red Hat, VA Software, Progress Software, and CollabNet. Many of these mid-size companies provides open source software development products and services to the large companies listed here. A number of small and start-up companies are described, including those like ActiveState, JBoss, Jabber and Compiere which have become global market leaders in their respective market niches within the past year by relying on the development and deployment of open source software products and services. The global success of these companies suggests careful study and consideration of strategic investment activities are timely matters to pursue. This motivates the case study that follows.

The case study examines GNUenterprise.org, an international virtual enterprise focused on the development and deployment of free open source software system framework for ERP and EB. The framework supports the development and integration of information system components for customer relationship management, corporate financial systems, and other business-to-business system modules. The study uses a Web-based approach to provide the data that empirically grounds this resource-based view of an organizational system. These data and the accompanying analysis help to what's involved in developing and deploying open source EC or EB capabilities, how it works; and the conditions that shape success or failure. The description, analysis, discussion, and implications of this case study are the result of this investigation.

Overall, this study reveals that their appears to be a variety of compelling business reasons for companies large and small to consider making strategic investments in free open source software development products, service, and processes that support EC, EB, and ERP applications.

Table of Contents

Executive Summary	2
Introduction	5
Research Questions	6
Background Concepts.....	6
What is Electronic Commerce?	6
What is Free, Open Source Software Development?.....	8
Open Source Software Development Practices	9
Open Source Software Products.....	11
Open Source Processes	11
Open Source Support Environments	12
Why is OSSD interesting?	13
What kinds of enterprises are interested in OSSD?.....	14
Large companies.....	15
Small and Midsize companies	19
Small, entry-level companies.....	19
Other enterprises and agencies interested in OSSD.....	21
What kind of enterprises could/should be interested in OSSD?	21
Case Study: <i>GNUenterprise.org</i> and the development of free open source ERP software.....	23
Computing infrastructure resources.....	30
Community communication resources	31
Scenarios of usage resources	31
How-to guides, To-do lists, FAQs, and other itemized list resources	31
Traditional system documentation resources.....	38
External publication resources.....	38
Open Web site and software resources	38
Software extension and customization resources	42
Free software license resources	44
GNUenterprise.org software development capabilities	47
The GNUenterprise.org software development process.....	47
The GNUenterprise.org community development process	49
The GNUenterprise.org community as a capability enabling Open EC/B	51
Analyzing the GNUenterprise.org Case	52
Economic conditions supporting free open source software development.....	53
Resources and Capabilities for Open EC/B	54
Personal software development tools and networking support.....	54
Beliefs supporting Free OSSD	55
Discretionary time and effort of developers.....	56
Trust and social accountability mechanisms.....	57
Free open source software development informalisms	58
Free OSSD capability enabling free, open ERP and EB systems	59
Revisiting the Research Questions	59
Implications and Enterprise Process (Re)Design Heuristics	61
Conclusions	63
References	65

List of Figures

Figure 1. A screenshot of a customer interface to an opaque, closed source E-Commerce process for purchasing goods from an electronic catalog at Amazon.com.	7
Figure 2. An architectural overview of an example open software, network, and hardware system, where each component (rectangle) and connector (oval) is available in an open source form.	10
Figure 3. A screenshot of the Java Community Process definition, sponsored by SUN Microsystems Inc.	17
Figure 4. An internal corporate portal for open source software behind the firewall (called "corporate source") at Hewlett-Packard.	18
Figure 5. Display of background information on the GNUenterprise.org and its GNUe software	24
Figure 6. A view of the history of the development of GNUe software.....	25
Figure 7. A public message from a GNUe project participant indicating her basis of financial support as a professional consultant and developer of Free software	26
Figure 8. A partial display of international companies providing resources and sponsoring the development of the GNUe software.....	27
Figure 9. Overview of the GNUe software system architecture and modules.....	28
Figure 10. A view of a public email thread transcript communicating project information and evaluative statements between project participants.....	32
Figure 11. A public transcript that records Internet-based chat in a real-time online project meeting.....	33
Figure 12. A snapshot of a community digest summarizes communications and activities across the GNUe project, with embedded Web links to source documents.....	34
Figure 13. A display view of a news item of interest to GNUenterprise.org participants	35
Figure 14. The GNUe Enterprise Designer, an interactive development environment (IDE) displaying a usage scenario for developers of GNUe-based applications	36
Figure 15. A screen display of links to an itemized list of "Frequently Asked Questions"	37
Figure 16. A view of a list of online user documentation found at GNUenterprise.org...	39
Figure 17. A display of a view into a source code Web directory associated with a GNUe software distribution version.....	40
Figure 18. A display of the Web site for acquiring and discussing a free, open source software content management system that supports the GNUenterprise.org Web site	41
Figure 19. Display showing a view of the SourceForge entry for GNUenterprise.org	43
Figure 20. A software bug report and status information maintained in DCL that is associated with a GNUe software component.....	45
Figure 21. The community property and copyright license that covers all GNUe software	46

Introduction

Many companies face a problem in determining how to best adopt and deploy emerging capabilities for E-Commerce and E-Business services. For example, interest in Web-based capabilities for the procurement of materials, repairs, and operations products, or for the acquisition of contracted services (e.g., consulting, facilities maintenance, and logistics) remains high, but most companies have not found substantial returns on their investment in these areas. When faced with the emergence of opportunities associated with wireless/mobile commerce, peer-to-peer (p2p) collaboration services, open source software, or the redesign of internal business operations to support E-Business, decision-makers face much uncertainty about how best to proceed, if at all. Simply acquiring and installing a Web-based p2p system for an internal or external business operations does not yield significant results. Transforming and realigning business processes, organizational strategies, and end-user work practices is required, but where and how does this occur? Overall, there is a basic and recurring lack of knowledge for how to determine and align the most effective choices for information system technologies, business strategies, and organizational transformation.

In a recent case study, [Scacchi 2001] reports how a federal government agency achieved a reduction in procurement cycle times of 20X and an annual operational savings of \$10M-15M. These results came from the transformation and realignment of information systems, business processes, corporate strategy, and work practices in the agency within a 1-2 year timeframe to support the agency's E-Commerce and E-Business initiatives. The results also came from a research approach where the research team and agency participants jointly developed and refined an open source computational models of the new business processes, resource dependencies, and work practices that together embodied the new organizational system that was sought [Noll and Scacchi 2001, Scacchi 2001].

The objective is to conduct a case study that examines how a firm can understand and transform its organizational systems to support an E-Commerce or E-Business initiative using free open source software resources or capabilities. Such a system may focus, for example, on back office activities associated with corporate financial operations, or on front office activities associated with customer relationship management. Alternatively, the focus may be directed as an organizational system where wireless, mobile, or p2p capabilities are sought.

The study employs a resource-based view of the organizational system involved in developing or adoption an open source EC/EB capability. The results of this arise from an analysis of the case study that uses Web-based snapshots of organizational information resources as data. This case study examines the GNUenterprise.org project. This study serves as a point of departure to explicate the concept of Open EC/B introduced in this report. Open EC/B results from combining open source software development with EC and EB.

Research Questions

This section introduces the research questions that are examined through an examination of the concepts, processes, and practices that give rise to open source software development (OSSD) projects and to EC/EB applications. To narrow our focus of the broad subject of EC/EB, the study examines the development of ERP systems for deployment and use within an enterprise. This background helps to set the stage for the study of GNUenterprise.org case, which focuses on the OSSD of an ERP system, and related EC/EB system modules.

Four research questions may explain how open source software development and EC/EB approaches might be brought together, and to what ends.

- How are OSSD processes similar to and different from those for EC or EB?
- What is involved in designing EC or EB processes or organizational systems to take advantage of OSSD processes, practices, and infrastructure?
- What kinds of challenges can make the transition from EC/EB to Open EC/B problematic, and how might these problems be mitigated?
- What kinds of resources or business capabilities are needed to help make Open EC/B efforts more likely to succeed?

In order to begin to understand and develop answers to these questions, the next section presents a background for what E-Commerce is, and what OSSD is. It also provides a brief survey of companies currently making strategic investments in OSSD that support business initiatives in EC or EB.

Background Concepts

In order to address the questions in the preceding section, we first provide some background descriptions of the concepts that underlie that study and analysis presented in this report. These focus on EC and Free OSSD.

What is Electronic Commerce?

Electronic Commerce is generally focussed on information systems, resources, and other IT related capabilities for buying and selling of products or services using the Internet or World-Wide Web (henceforth, "Web") as the medium or venue for this commerce.

Electronic Business generally follows a similar form and purpose, but with the emphasis on internal business processes within a single unit, multi-unit, or multi-site enterprise. Both EC and EB are active areas of commercial development and deployment by many large software system vendors and systems integration consultancies.

A common example can be found in the business-to-consumer form of EC typically experienced at Amazon.com. Figure 1 presents a view of an online shopping experience where a user has browsed Amazon's online product catalog, selected the items displayed, and is ready to proceed forward in completing an online order placement (purchase) transaction. The EB process for purchase ordering is opaque (and it's "1-Click" patented), but perhaps implicit to the customer in this EC transaction, thanks to automatically generated email messages about an order's status and fulfillment progress.



Figure 1. A screenshot of a customer interface to an opaque, closed source E-Commerce process for purchasing goods from an electronic catalog at Amazon.com.

It is likely that the EC and EB processes underlying this transaction and the other EB processes related to it (e.g., order fulfillment, inventory control, and distribution logistics) are coded within software programs. The business process logic in these programs is accessible only to Amazon programmers, rather than to its customers or internal operations users. In other words, Amazon's EC and EB processes exist only in closed, inaccessible form.

Some approaches to EC and EB seek to be very comprehensive in their attempt to develop, deploy, and/or integrate information systems for corporate financial operations (e.g., general ledger, accounts payable and receivable, etc.), sales and support (contact management, customer relationship management), and supply chain management (inventory control, demand forecasting, supplier relationship management, etc.). Enterprise Resource Planning (ERP) systems were developed and deployed in the 1990's as the IT approach to prototyping and ultimately realizing comprehensive EB processes or internal operations [Curran and Ladd 2000, Keller and Teufel 1998, Skok and Legge 2002]. These efforts emphasize selection, customization, and deployment of EC or EB system modules for business operations relevant to an enterprise's incremental and iterative transition to EC/EB [Keller and Teufel 1998].

To succeed in an EC/EB initiative, an enterprise must provide and support a networked computing infrastructure and resources to customize and deploy the system modules. It must do so in a manner that is compatible with overall business objectives, labor and market conditions that ensure the sustained viability and longevity of the deployed EC/EB system. Unfortunately, the overall costs of acquiring and deploying a comprehensive ERP system that enables both EB and EC processes, often runs well into millions of dollars (sometimes as high as one hundred million dollars) and years to deploy. These kind of ERP solutions are limited to the largest national or international enterprises. Subsequently, such a situation puts the capability for ERP as a basis for EB and EC well beyond the means and timeframes that are available to the large numbers of small or midsize enterprises. Therefore, the opportunity exists to understand what alternatives, if any, might be available for developing and deploying ERP systems that enable EB or EC initiatives for enterprises of different sizes. Such an alternative may be found within the emerging OSSD movement.

What is Free, Open Source Software Development?

OSSD represents a relatively new approach to the development of complex software systems [Feller and Fitzgerald 2002]. OSSD generally relies on a global community of software developers and users who seek faster, better, and cheaper alternatives to closed proprietary systems. In the past year or so, EC and EB systems have begun to be developed using OSSD techniques.

In most OSSD situations, the resulting software system and its associated Web-based documents or development artifacts are globally accessible at little or no direct cost. Furthermore, a defining requirement of OSS is how their intellectual property rights are assigned and protected. The terms and conditions of the copyright or license associated with OSS typically assert the following kinds of property rights or "freedoms" to anyone

who seeks to employ or use the software [cf. DiBona, Ockman, and Stone 1999, Pavlicek 2000, Williams 2002]:

- Freedom to run the program for any purpose;
- Freedom to study how the program works and adapt it to their needs;
- Freedom to redistribute copies of the software at will;
- Freedom to improve the OSS program and to distribute the altered version;
- Required distribution of the originating license that specifies the freedoms and rights concerning the preceding properties.

These rights or freedoms do not prohibit charging fees to access or acquire OSS, though typically there are no direct cost for the OSS itself. Instead there may be costs associated with acquiring the media (e.g., CDROM, books) through which the OSS is distributed. Similarly, the rights or freedoms do not restrict who may provide support, system integration, or consulting services for the OSS. This is in contrast to the strictly controlled provision of support or services offered for closed source, proprietary software systems. These rights and freedoms stand in marked contrast to those offered with the selection, customization, and deployment of commercial ERP and other systems for EB or EC.

In the following sections, we look more closely at the practices, products, processes, and support environments that help make OSSD an interesting alternative for EC/EB. This is help us better understand how OSSD and EC/EB might come together to create alternatives that embrace the rights and freedoms of OSS with the potentially strategic information system capabilities and resources that enable an ERP system.

Open Source Software Development Practices

Open source is not the same concept as "open systems". Open source is a broader, more encompassing technique for exposing access to the underlying functionality, operation, or interoperation of a software system. Open systems traditionally refer to a technology scheme that provides customers, external developers, or end-users to access the internal functions of a complex system via "public interfaces." These interfaces take the form of accessible connectors or plug sockets. The structure of these interfaces denote the points of contact through which pre-specified types of program data, control signals, and error messages flow in or out. Otherwise, the modules or components to which the connectors are affixed are opaque, closed black boxes.

In open source software (OSS), the source code, as well as its surrounding documents and artifacts, all serve as the public interface to the system. Access to system functionality is not limited to functions calls through "application programming interfaces" (APIs). Access to functionality, as well as the ability to enhance, restructure, tune, debug, or re-host system functionality is realized through access to open source code, documents, and artifacts. An open system may consist of hardware, software, and network system components. Subsequently, the potential exists for making all of the components functionality accessible, transparent, and open through public interfaces that consist of the components "source code", documents, and artifacts. Figure 2 provides a view of an open system composed out of open source hardware, software, and network components.

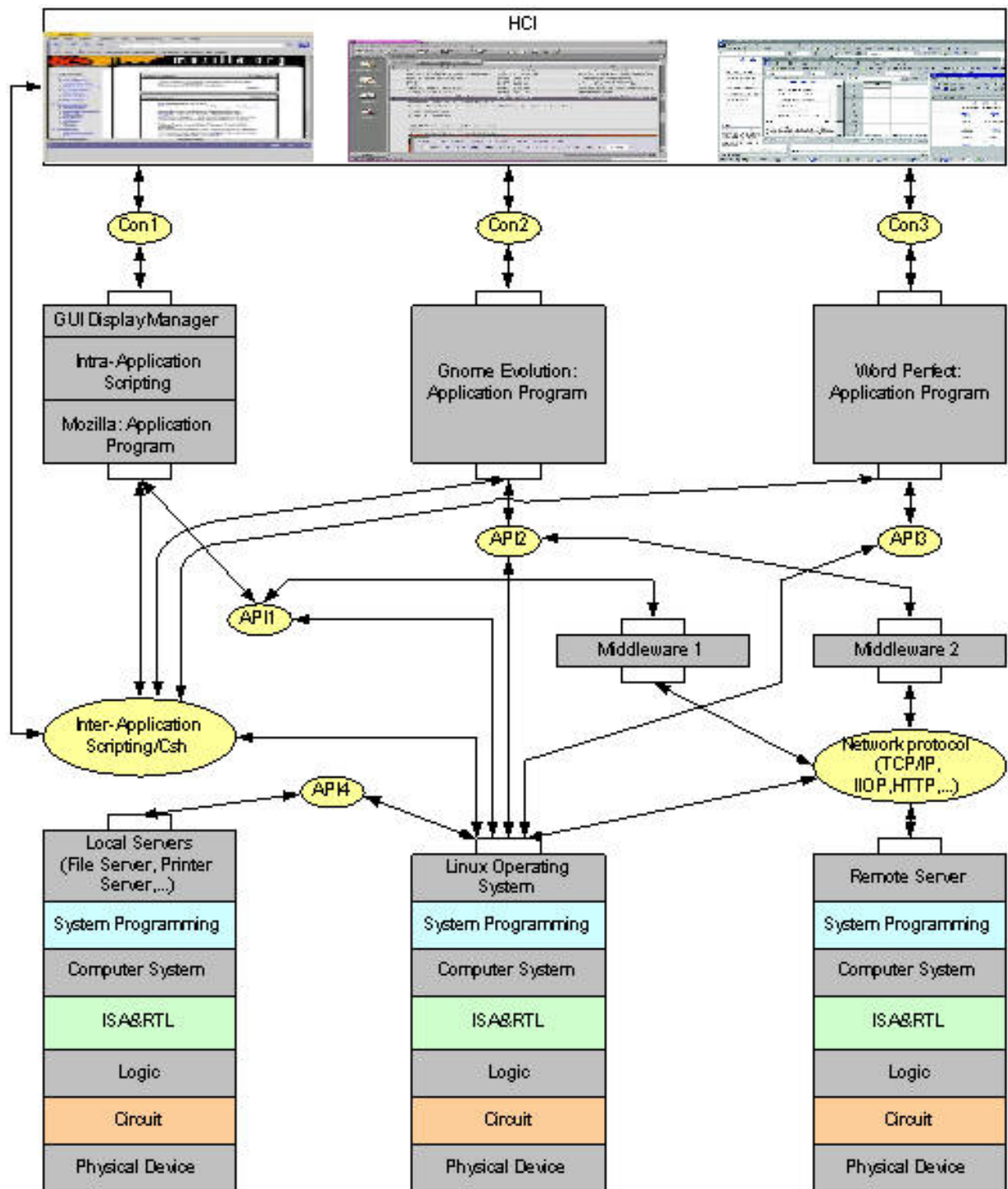


Figure 2. An architectural overview of an example open software, network, and hardware system, where each component (rectangle) and connector (oval) is available in an open source form. (Diagram by Xiaobin Li, May 2002) [Li and Scacchi 2002].

With this in mind, we now turn to examine the products, processes, and support environments for OSS.

Open Source Software Products

OSS program code is the typical focus of most OSSD activities. These computer programs are written in a programming language like C, C++, Perl, Python, and others. Documents that specify or describe how these programs function or interoperate are also products of open source software development. These documents may include specification or design diagrams, end-user manuals, program installation scripts, threaded email discussion forums, Web-based source code repositories, and other Web site contents. OSSD projects rely on a diversity of software *informalisms* [Scacchi 2002b] as information resources, documents, artifacts, or products that can be browsed, cross-linked, and updated on demand. These informalisms are socially lightweight information structures for managing, communicating, and coordinating globally dispersed knowledge about who did what, why, and how. These informalisms are easy to learn and use as semi-structured representations that capture software requirements, system design, and design rationale. As OSS developers are themselves end-users of their systems, then software system requirements and design take less time to articulate and negotiate, compared to software engineering projects that must elicit requirements and validate system design with end-users who are generally not software system specialists. Thus, a lesson learned from these observations is that practitioners of OSSD should be both users and developers of open source practices, and these practitioners should be provided with the ability to easily create, modify, share, and discuss informal descriptions of OSSD products, practices and outcomes.

OSS concepts can apply to any product that can be produced with or through the use of computing systems that can be networked together. As seen next, these products can also include the source code to the processes for technical work or business operations, if these processes can be specified in a form that can be compiled or interpreted for automated or interactive execution in a networked computing environment.

Open Source Processes

OSS programs emerge as the result of technical activities that are arranged and ordered in a manner that can be described as a process—an OSSD process. This process may be *ad hoc*, difficult to describe and repeat, or it could be more structured and follow a pre-articulated scheme or formal process. In any case, it is reasonable to describe how the OSS products are developed, used, and evolved as processes. Furthermore, these processes may be descriptive, proscriptive, or prescriptive. Descriptive processes describe *what previously occurred*, by whom, when, where, etc. Proscriptive processes describe *what could be done* at certain points or in response to some event or condition. Prescriptive processes describe how software development activities *should be done* when supported by some means or mechanism that checks for compliance with the process prescription. In any of these forms, processes can be codified into computational models that can be analyzed, simulated, and iteratively refined by process users in many

different ways [Noll and Scacchi 2001, Scacchi 2000, 2002a, Scacchi and Mi 1997, Scacchi and Noll 1997].

OSSD projects can enact "Internet time" development practices, much like Microsoft, Netscape, and others [Cusumano 1999, MacCormack 2001] follow when market conditions demand rapid response to substantial competitive threats. Internet time efforts emphasize minimizing time to market and delivery of incremental improvements in functionality, instead of complete well-engineered functionality that gives rise to much less frequent product releases. Internet time development depends on collecting feedback from end-users as a way to determine which incremental functionality, and which perceived errors in available functionality matters most.

OSSD projects are iteratively developed, incrementally released, reviewed and refined by OSS developers working as peers in an ongoing agile manner [cf. Cockburn 2001]. These methods ensure acceptable levels of quality, coherence, and security of system-wide software via continuous distributed peer review, testing and profiling. OSSD efforts are hosted within decentralized communities of peers [Kogut and Metieu 2001, Scacchi 2001, 2002, Sharman 2002] that are interconnected via Web sites and OSS repositories. Community oriented OSSD gives rise to new kinds of requirements for community building, community software, and community information sharing systems (Web site and interlinked communication channels for email, forums, and chat). In contrast, most system engineering projects associated with major software development efforts are targeted for a centralized corporate setting, where access and visibility may be restricted to local participants. OSSD standards [Freericks 2001] are apparently easier to access and follow due to their Web-based deployment, and a long history of community oriented participation in developing implementation-oriented standards in an open source manner. These compare favorably to the institutionally oriented processes used to develop software engineering standards that are much more cumbersome and often less effective at ensuring system quality [Scacchi 2002].

Open Source Support Environments

OSS emerges from the efforts of developers who are distributed across space and time. They do not work in a single or central workplace, and often there is no formal management hierarchy in place to schedule, plan, and coordinate who does what, with what resources, etc. Instead open source developers contribute their effort to projects that they find interesting, significant, or otherwise professionally compelling. OSS developers generally have regular paid jobs, though they may or may not be paid to work on an open source project. Thus, traditional organizational models for how to motivate employees or how to organize and manage technical staff may not apply. Nonetheless, open source development projects thrive, as it now appears that tens of thousands of OSSD projects are underway.¹

¹ For example, the Web portal site, www.sourceforge.net, identifies more than 40,000 registered open source development projects and more than 400,000 open source developers. 15% of these projects are identified by their developers as "stable" systems suitable for production application, or "mature" systems being sustained and incrementally evolved to improve their usability, system performance, and to expand the diversity of platforms on which they operate.

OSSD projects are "organized" as a loosely knit community of interested developers and end-users who work and interact online via Web-based computing environments [Scacchi 2002b]. These environments provide access to a global information infrastructure that includes routine support for Email and electronic bulletin board, and Web sites for posting or sharing open source artifacts. They also provide public access to centrally administered multi-version source code directories, software extension schemes and mechanisms (e.g., multi-application scripting languages, like Perl and Python, to enable interoperating systems), and more [Scacchi 2002]. Developing trust, "geek fame", and being recognized by peers for technical contributions [Pavlicek 2000] are part of the "glue" that binds open source developers together with their global information infrastructure to create the productive units or virtual organizations [Noll and Scacchi 1999] that populate the world of OSSD. These virtual organizations are thus part of what must be reproduced and enacted in the world of open government and open acquisition.

OSSD tools are inexpensive/free and comparatively easy to use and learn. They are both given and received as public goods or gifts to the community [Bergquist 2001]. The most widely used OSS tools support concurrent version control and repository management [Fogel 1999], Web servers and browsers, communication applications (threaded email discussion forums, instant messaging), bug/issue reporting and resolution tracking, and various code development tools (text editors, integrated development environments, etc.). Access to and availability of OSS tools is generally not a problem or barrier to participation in an OSSD project.

Faster and better OSSD conditions in tend to drive down the cost of developing software in terms of schedule and budget resources. Most OSSD projects are voluntarily staffed by people who want to work on the project, who will potentially commit their own time and effort, and who find personal and professional benefit from the OSSD development efforts [Scacchi 2002b]. Minimal management or governance forms [Sharman, *et al.*, 2002] are used to direct OSSD efforts, compared to the more rigid hierarchically managed, planned, staffed, controlled, and budgeted project activities typical for traditional development efforts associated with corporate initiatives for EC or EB .

Why is OSSD interesting?

Interest in OSSD arises from its combination with tools, techniques, and concepts that support EC and EB. Traditionally, the development and deployment of EC/EB IT solutions is considered a costly undertaking that may take years to realize a return on investment. OSSD begins to erode the legacy of cost and time to deploy in EC/EB application in a number of ways.

First, OSSD appears to offer lower cost of development, customization, and ownership. Development costs are driven down through access and use of free/low cost tools, development tools and Web-based deployment environments. These can include core EC/EB application modules for enterprise financials, sales, and supply chain management. Customization and ownership costs may decline due to non-exclusive access to the source code that constitutes the EC/EB application modules. Traditionally,

source code for these modules is only available from proprietary vendors that offer monopolistic product support contracts to their customers.

Second, OSSD may offer higher quality products and outcomes due to competitive forces. Consultants and system integrators cannot rely on exclusive monopolistic service contracts to sustain their business relationship with their customers, nor can they maintain exclusive access to the source code of the EC/EB application modules. Further, these application modules are subject to an open, public peer review of incremental contributions, as well as discover and discussion of problems or bugs in currently deployed modules. This means customers can benefit from open community-oriented problem diagnosis and resolution.

Third, OSSD offers faster turnaround in the resolution of problematic features, gaps, or security holes found in OS software, compared to the slow and periodic nature of problem resolution releases that proprietary vendors traditionally offer. This is particularly true of widely deployed OSS systems that are continuously evolved in Internet time, such as those that constitute the predominant software infrastructure of the Web, and thus Web-based EC and EB.

Last, OSSD has been subject to a recent influx of empirical research studies that substantiate, challenge, or extend the "faster, better, and cheaper" characteristics of a small but growing number of OSSD projects [Scacchi 2002c]. OSSD is emerging as a hot research topic, and major OSSD projects associated with the ongoing development and expansion of the Internet and World-Wide Web information infrastructure, have begun to appear. Studies have been reported on the Apache Web server [Mockus, Fielding, Herbsleb, 2000, 2002], the Mozilla Web browser [Reis and Fortes, 2002], the GNU/Linux operating system kernel the [Schach *et al.*, 2002], the GNOME User Interface [Koch and Schneider 2001; German 2002], the FreeBSD operating system [Jorgensen, 2001], and the deployment of OSSD tools and techniques "behind the corporate firewall" within Hewlett-Packard corporation [Dinkelacker, Garg, Miller, Nelson, 2002].

Therefore, this is a most opportune time to undertake study of how OSSD may be applied to support the development of EC/EB applications. Following this, the next section examines what enterprises are interested in making strategic investments in OSSD.

What kinds of enterprises are interested in OSSD?

Companies of all sizes within the IT industry in general, and software industry in particular have begun to make strategic investments in OSSD projects. Who are these companies? Why are these enterprises interested in OSSD, and why have they made strategic investments? How do they manifest their interest in OSSD? What other kinds of organizations are interested in OSSD as a strategic investment? These are the kinds of questions examined in this section.

Large companies

There is a small but growing number of large companies² in the IT industry that are investing in OSSD in a publicly visible manner. Perhaps we should not be surprised to find that most of the major IT hardware system vendors, including IBM, SUN, HP (and now HP-Compaq), Apple, and major Internet Service Providers like AOL, have made investments in OSSD efforts that support new products or service offerings.

IBM, SUN, and HP among others have invested in supporting the porting of the open source GNU/Linux operating system to a variety of their hardware product lines, from mainframe computers at IBM to hand-held pocket PCs at HP-Compaq. Other IT software companies like Oracle and SAP have also invested in re-hosting their premier enterprise-class products on GNU/Linux operating systems. Popular EC Web sites like Amazon and Yahoo host their portal, EC, and EB applications on open source systems, primarily as an operational IT cost-reduction technique.

Apple Computer in 2001 released its latest operating system, OS X, for its Macintosh computers. It is based on an open source operating system kernel (called *Darwin*) which is similar to the FreeBSD operating system [Jorgensen 2001].

SAP joined the open source world in 2000 by relicensing its main database management system product, SAP DB, using the GNU Public License, and the establishment of a community Web site for it at www.sapdb.org. Though the SAP DB was previously marketed as the host DBMS for the SAP R/3 ERP system, SAP was largely unsuccessful against competitors like Oracle and Microsoft, who already had a DBMS installed in SAP ERP customer sites. Nonetheless, SAP DB does appear to be a full-featured, enterprise-scale DBMS product that is offered as an open source software product. However, it should be noted that when SAP DB is used in conjunction with SAP R/3 ERP software, SAP requires a contracted product support license for SAP DB. Not surprisingly, few OSS developers seem to embrace or develop their open software systems that require a DBMS by using SAP DB. Similarly, it seems unlikely that a large enterprise customer for an SAP R/3 ERP system is likely to be motivated to adopt an open source DBMS, while at the same time adopt a closed source ERP system that runs on it, since most SAP R/3 deployments represent multi-million dollar installation and support contracts from a single provider [Keller and Tufel 1998]. Thus, the strategy that SAP seems to be pursuing with SAP DB is to offer a lower acquisition cost to its SAP R/3 customers in an attempt to dissuade them from purchasing higher cost DBMS products from Oracle, Microsoft, or other enterprise DBMS vendors (e.g., IBM). It also appears that such an open source product pricing strategy for SAP DB, and its close affiliation with SAP R/3 prevents or mitigates the chance for a switch to a competitive ERP system occurring in a customer's EB operations.

AOL, which previously acquired Netscape Corporation, has invested in supporting the ongoing development of a free, open source Web browser called Mozilla (cf. www.mozilla.org). Mozilla was recently released for public consumption, featured in an

² "Large" here refers to companies with multi-billion dollar annual revenues.

article in *Time* magazine, and is the frequently mentioned in rumors spread over the Internet as the basis for the next AOL (8.0) Web browser.

These investments may be seen as a strategic move that might prevent, mitigate, or diffuse Microsoft's growing penetration and dominance of the computer operating system market. However, these large companies have made other investments that are not necessarily aimed at Microsoft. For example, IBM has established its *developerWorks* Web site for new OSSD products, and its high revenue, highly profitable Global Services division offers consulting services for the deployment and integration of OSS products with legacy enterprise applications. SUN has established its position in supporting OSSD based on its Java technology product line. This is most visible in SUN's Java Community Process, shown in Figure 2. HP-Compaq has invested in, demonstrated ongoing success and expanding commitment in a hybrid approach to OSSD they call, "progressive open source" [Dinkelacker, Garg, Miller, Nelson 2002]. Progressive open source is a technique for deploying a collaborative software development environment that can support external software product or internal application development behind HP's corporate firewall, but across a global corporate environment. A view from behind the corporate firewall of HP's in-house OSSD effort is displayed in Figure 3.

Other large companies not in the IT industry are also investing in OSSD to support their EC/EB initiatives. Examples are found in the international business banking industry. Dresdner Kleinwort Wasserstein (DKW) developed and made free, the *openadaptor*TM software technology, which is an open source middleware mechanism for expediting the connection of disparate EB systems found in the investment banking industry. DKW has established a sub-unit called *openadaptor.org* to handle ongoing development, community building, licensing issues, and partnership alliances associated with openadaptor. Elsewhere, Barclays Global Investors (BGI) sought to maximize the benefits of global collaboration through the use of a Web-based software development framework from CollabNet called *SourceCast*TM. With development groups based in numerous locations including London, San Francisco, Sydney, Tokyo, and Toronto, BGI sought to achieve an end-to-end, integrated and consistent approach across its entire global software development practice. BGI selected SourceCast as the basis for its collaborative software development infrastructure for more than 150 internal system development projects ranging from high-end equity trading systems and global XML development to Web-based client order systems.³

Overall, it does not appear that the experience or efforts of DKW or BGI are unique to the international banking industry. Instead, it is probably coincidental that two banking institutions would embrace and publicize their efforts in OSSD. Nonetheless, it seems likely that other large non-IT industry companies might soon begin to invest in OSSD in support of their EC/EB initiatives. We suspect this may be particularly true for those companies that are involved either in the integration of complex multi-vendor enterprise applications like DKW, or those that entail providing a common collaborative development environment that can be readily deployed across globally distributed business units, much like BGI.

³ Source: http://www.collab.net/solutions/networks/enterprise/solutions_at_work.html/#bgi, June 2002.

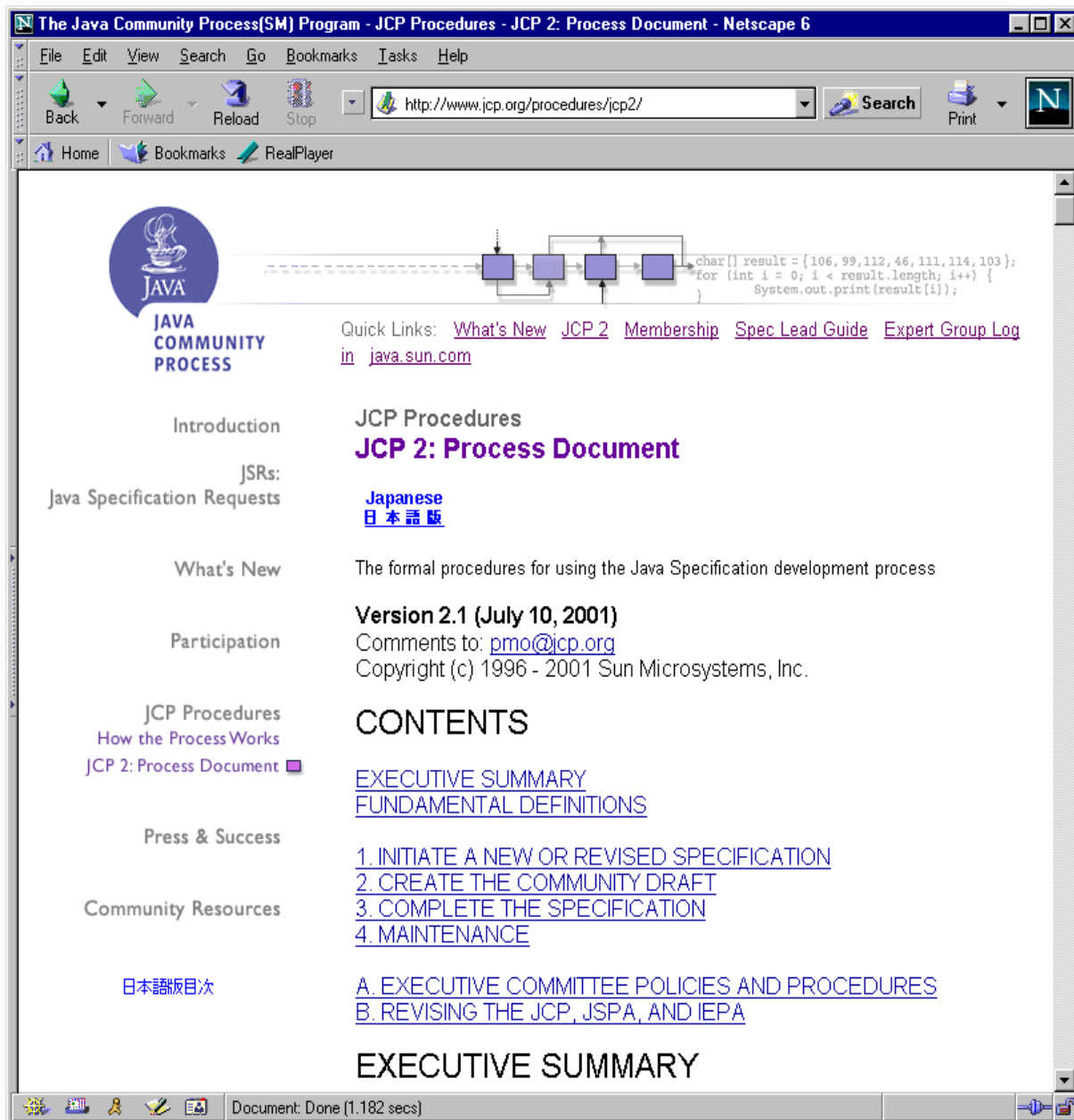


Figure 3. A screenshot of the Java Community Process definition, sponsored by SUN Microsystems Inc. (Source: <http://www.jcp.org/procedures/jcp2/> April 2002).

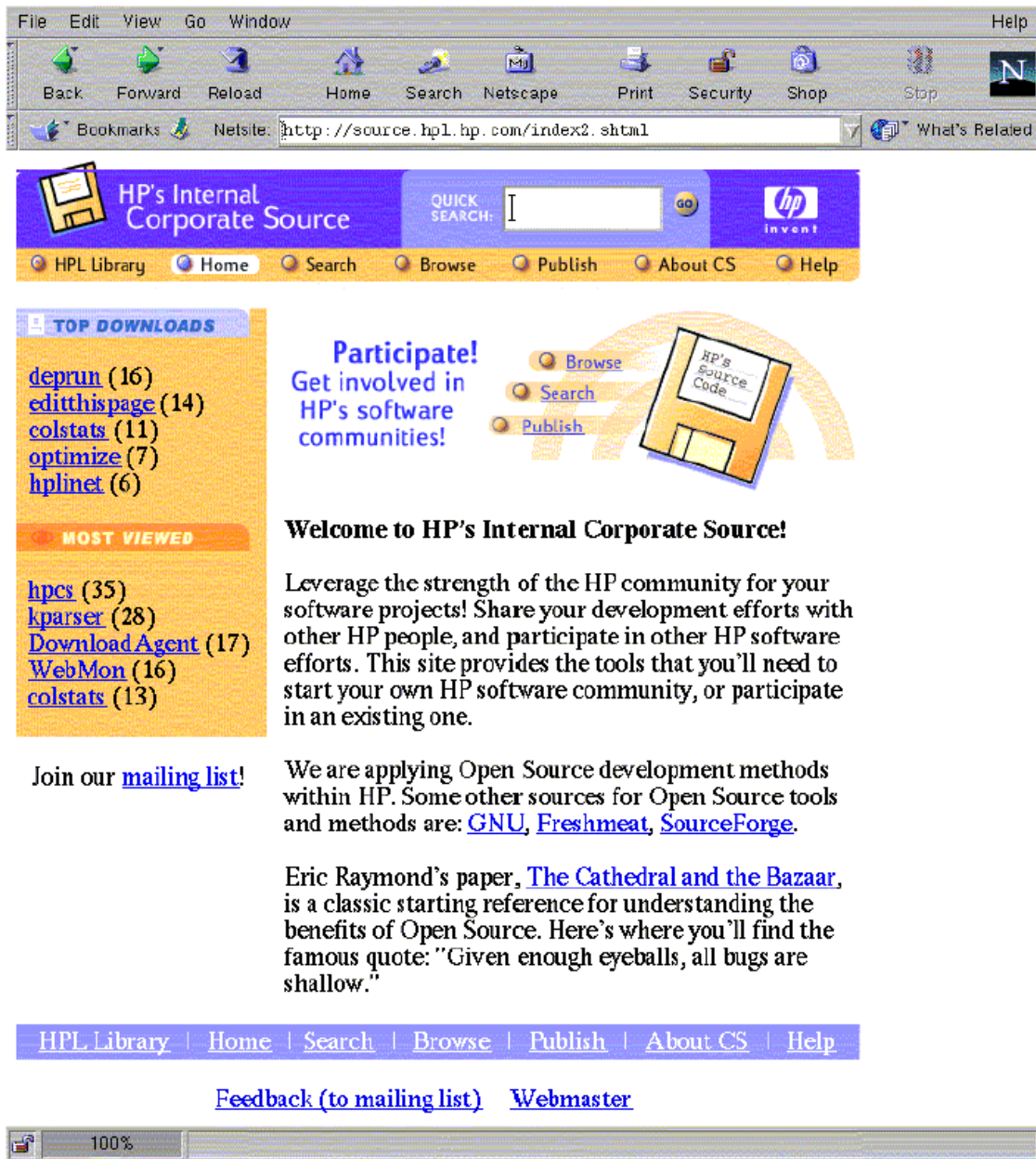


Figure 4. An internal corporate portal for open source software behind the firewall (called "corporate source") at Hewlett-Packard (Source: Dinkelacker, Garg, *et. al.* 2002).

Small and Midsize companies

Red Hat, VA Software (formerly VA Linux), and Collab.Net are probably the most visible of companies whose products or service offerings are primarily based on OSSD efforts. Early leaders in the OSSD movement in the late 1990's founded these companies.

These companies for the most part have survived the economic downturn that has affected most IT companies in the past few years, as well as surviving rapid rise and fall in their public valuations, as part of the dot-com industry shakeout. In sustaining business operations, these companies have been able to migrate away from unprofitable business operations (e.g., VA Linux's exit from the Linux workstation business) and more towards revenue generating operations associated with support and consulting services (including custom systems integration engagements). But these are not the only small to midsize businesses to sustain and grow value through OSSD based product or service offerings. For example, Progress Software, the world's leading vendor of embedded database management systems, and #2 vendor of E-Business integration software behind SAP, is another company that has made strategic investments in OSSD. Progress recently acquired NuSphere Inc., which is the world's leading provider of the MySQL relational database management system. MySQL is now included with most GNU/Linux operating system distributions, so that there are on the order of 1M copies of MySQL already distributed. Thus, there are small and midsize companies with sustained operations and revenue generation focused on the development/support of OSS products and services.

Small, entry-level companies

In general, there are two kinds of small, entry-level companies that are interested in OSSD. First, there are entry-level start-up firms and established small companies that seek to acquire, use, or otherwise develop OSS as the basis for the products or services they intend to offer for generating revenue. ActiveState and Zope would be examples of sustained but growing small companies of this kind. Second, there are entrepreneurial system integrators and consultancies that seek to sell/market their support services through offering free or low-cost open source software based solutions. Companies of the second kind are often those of the first kind as well, though there are examples of many small companies of each distinct kind. To simplify matters, here we choose to examine a small sample of those companies that act as both kinds.

Compiere, based in Europe, has become the world's largest provider of ERP systems, with more than 250K copies of its open source software already distributed (i.e., downloaded from its Web site) around the world. Traditional leaders in the ERP market like SAP, Oracle, Baan, JD Edwards, and PeopleSoft together claim no more than 20-25K copies installed of their costly, closed source ERP software systems. Compiere is not a direct competitor to the large ERP software vendors, unless or until those companies undertake product pricing and sales initiatives that are targeted to the small businesses that Compiere hopes to entice with its offerings. Compiere is a competitor to the GNUenterprise.org project examined in this study. However, Compiere relies on the use of proprietary, closed source relational database management system (from Oracle). So

for many who acquire a copy of the Compiere ERP software, it mainly serves as a demonstration or evaluation system whose usefulness is determined by the proprietary license from Oracle for their DBMS product (which is available at no cost for a limited period). Whether Compiere could readily rehost its open source ERP software onto the open source SAP DB and thereby realize a more complete open source ERP software environment is unclear. However, such a move would potentially put SAP in a difficult competitive position, since Compiere might then be able to have some effect on the shape of the ongoing development of enterprise-scale ERP software systems due to its comparative mindshare advantage in the OSSD community and the hundreds of thousands of copies already distributed. Thus, Compiere may offer an open source ERP system, but not a free or open source DBMS that can support it.

JBoss is the world's leading provider of Java-based application servers and server frameworks, with several hundreds of thousands of copies having been distributed. This compares to the estimated total of 10-30K copies of proprietary closed source software products of this kind offered by IBM (WebSphere), BEA Systems (WebLogic) and others. The JBoss business model is centered around service offerings for consulting, custom systems integration, and training. It recently began to train and certify independent JBoss consultants from within the JBoss.org community, in order to help meet its rapidly growing customer service base. JBoss software is free and open source, and is usable in full once properly installed.

Jabber was perhaps the first company to offer open source software for instant messaging (IM) systems. Jabber appears to compete with IM products from AOL, Microsoft, Yahoo, and others. However, Jabber's business strategy is to seek compatibility and interoperation with existing proprietary IM products. Jabber's IM software surrounds and dissolves the differences and incompatibilities⁴ that exist among the competing closed source IM products through an alternative system architecture for IM that enables interoperation between and among proprietary closed-source IM systems. Thus users of Jabber software can send and receive instant messages or chat with proprietary IM clients, whereas proprietary IM clients can only send and receive messages from IM clients of the same one vendor. Jabber IM software is open source, and is usable in full once properly installed.

Ximian is the main firm backing the development of the GNOME graphic user interface (GUI) for the GNU/Linux system. GNOME is now bundled with nearly all GNU/Linux system distributions, thus realizing many millions of copies going into distribution. While these number of copies is minute compared to the existing global deployment of Microsoft's Windows interface, SUN has licensed GNOME to use henceforth as its standard GUI for distribution with its Solaris (SUN Unix) operating system. Ximian is also developing a C# compiler and .Net Web services that seek to replicate the functionality of those products being developed at Microsoft. GNOME software is free and open source, and is usable in full once properly installed.

⁴ These embody vendor "lock-in" strategies that are embedded in the closed source IM products.

Overall, each of these small entrepreneurial companies seeks to offer high performance OSS products at little or no direct cost, as a strategy for developing price competitive, sustained system integration and support services. For these companies, their OSS products might be seen as "loss leaders" that attract customers who are the target of service offerings. Distribution of thousands to millions of copies of different OSS applications and tools seems to insure an ongoing regional (or industry-specific), national, international, and global market demand for associated service offerings. Innovative entrepreneurial opportunities for entry-level companies may emerge to offer logically centralized support services, while customers may be physically distributed and organizationally decentralized. However, OSS companies in general do not hold a monopolistic, vendor lock-in position as sole provider of ongoing support services. Thus they must compete based on the convenience, speed, quality, and cost of their service offerings.

Other enterprises and agencies interested in OSSD

Two other kinds of enterprises that are increasing being associated with the emergence and deployment of OSSD projects are non-profit enterprises and government agencies. Non-profit foundations have been established to assume legal ownership (and defense) of the property rights of a small number of free or OS projects or community organizations. The Free Software Foundation was the first of these, associated with thousands of OSSD projects that employ the GNU Public License (GPL) and its variants (e.g., the Lesser/Library GPL), in order to insure that the software developed by these projects remains free and open in line with the terms and conditions stipulated in their licenses.⁵ The GNU/Linux operating system, which an estimated 10M copies in distribution, is the most visible of these projects that relies on the GPL to define, insure, and protect its community property rights, and thus to prevent them from being co-opted in corporate or individual private property. The Apache Software Foundation has a similar purpose for the Apache Web server, which operates a majority of Web servers on the Web.

National governments, particularly those in Europe and South America, have begun to put forward or debate public policies that encourage or mandate the adoption of OSS for use in national government agencies. The reasons for embracing such public policy are interesting and significant, but beyond the scope of this report, though they are beginning to be addressed in other reports [Nunez 2002, Scacchi 2002a] and proposed legislation [Cortiana 2002]. Nonetheless, it seems likely the OSSD will increasingly be viewed as a subject amenable for public policy within developed or developing countries.

What kind of enterprises could/should be interested in OSSD?

Based on the kinds of companies identified above, there is no *a priori* reason to expect that many other enterprises in different industries will not become increasingly interest in adopting OSSD projects for internal EB or external EC processes. Among those we expect to be early adopters of OSSD for EB or EC would likely include semiconductor, hardware component, or subsystem vendors like Intel, Conexant, Motorola, Texas Instruments, Nokia, Ericsson, Siemens and Broadcom. Intel and Ericsson have already

⁵ The GPL is constituted within U.S. copyright law frameworks. The European FSF also provides the GPL for use in European countries that might have different copyright frameworks.

invested in GNU/Linux projects, and thus represent the first in the semiconductor industry to begin to invest in this area.

Consumer electronics and appliance companies like Sony or General Electric are also likely adopters of OSSD projects. In particular, Sony's Computer Entertainment Group, makers of the Sony PlayStation 2 (PS/2) computer game console (with approximately 50M units sold) has recently released software development tools that enable OSSD projects that can run on and use the PS/2 console, via its embedded GNU/Linux operating system variant. This move by Sony may subsequently enable OSSD projects that seek to create *OSS games* for EC or EB, or that create an EC infrastructure that supports emerging global game community economies, like that for the virtual world of *Norrath*, which recently was identified as the 77th largest GDP economy in the world [Castronova 2001]. *Norrath's* economy⁶ is almost entirely electronic, relying on the global EC infrastructure for sales, auctioning, sourcing, payment, and reputation services provided by eBay. *Norrath* is a virtual world hosted within the *Everquest* computer game world operated by Sony and its subsidiaries.

Moving farther away, we also eventually expect to see other high tech firms in the network systems, automotive, aerospace, and biotech industries to be among those that will adopt OSSD projects that enable or support internal EB processes or external EC processes. Clearly this is speculation, but it seems possible that companies in these industries will seek OSS solutions when they incorporate software systems that are part of, but not the entire, a product or service offering. Thus, aerospace companies like Boeing, and automotive companies like Ford, will seek OSS-based system component solutions that will be embedded in the airplanes or cars they sell.⁷

Overall, perhaps one of the most surprising findings from the preceding survey of companies investing in OSS products and services, is the number of companies that have attained market leadership or significant contender positions. Clearly, these attainments are realized in terms of number of units (software systems) shipped/distributed, rather than on revenue generated. But this is still a significant accomplishment, given the ability of these firms to do so in direct competition to some of the largest and well-funded companies in the IT industry.

As such, we now turn to examine a case study of a global organization that seeks to combine the OSSD approach to develop and deploy EC/EB systems that are based on open source ERP system modules.

⁶ eBay based EC for *Norrath* enables trade in game resources among game players (*Norrath* residents) that must otherwise be acquired through many hours of game play.

⁷ Every airplane or automobile is a potential host for a mobile Web server and browser embedded with the individual vehicle, for vehicles networked within a fleet, or for vehicles serving/accessing content from the global Web.

Case Study: *GNUenterprise.org* and the development of free open source ERP software

GNUenterprise.org is an international virtual organization for software development [Crowston and Scozzi 2002, Noll and Scacchi 1999] based in the U.S. and Europe that is developing a *free, open source Enterprise Resource Planning (ERP) systems and related E-Business capabilities*. This organization is centered about the GNUenterprise.org Web portal and global Internet infrastructure that enables remote access and collaboration (see Figure 5). Developing the GNUe software occurs through the portal, which serves as a global information sharing workplace and collaborative software development environment. It's paid participants are sponsored by one or more of twelve companies spread across the U.S. and Europe. These companies provide salaried personnel, computing resources, and infrastructure that support this organization. However, many project participants support their participation through other means (cf. Figure 7). A partial view of these companies appears in Figure 8. In addition, there are also dozens of unpaid volunteers who make occasional contributions to the development, review, deployment, and ongoing support of this organization, and its software products and services. Finally, there are untold numbers of "free riders" who will simply download, browse, use, evaluate, deploy, or modify the GNUe software with little/no effort to contribute back to the GNUenterprise.org community [cf. Olson 1971].

GNUenterprise.org is community-oriented project, as are OSSD efforts [Scacchi 2001, Sharman *et al.*, 2002]. The project started in earnest in 2000 as the result of the merger of two smaller projects both seeking to develop a free software solution for EB applications. More information on the history of the GNUenterprise.org project can be found on their Web site, as suggested in Figure 6. The target audience for the GNUe software modules are envisioned as primarily small businesses that are underserved by the industry leaders in ERP software, perhaps due to the high cost or high prices that can be commanded for commercial ERP system installations. Many of these target companies might also be in smaller countries that lack a major IT industry presence.

GNUenterprise.org is a free software project affiliated with the Free Software Foundation and the European FSF. The ERP and EB software modules and overall system architecture (see Figure 9) are called the GNUe software. All the GNUe software is protected using the GNU Public License (GPL) [DiBona, Ockman, and Stone 1999, Pavlicek 2000, Williams 2002]. This stands in contrast to the ERP software from Compiere, which depends on the use of a commercial Oracle DBMS. Thus, GNUenterprise.org is better characterized as a free open source project, rather than simply an open source development project [Fitzgerald and Feller 2002].

GNUenterprise.org is not in business as a commercial enterprise that seeks to build products and/or offer services. It is not a dot-com business, but is a "dot-org" community venture. GNUenterprise.org is more of a pre-competitive alliance of companies and individuals that want to participate in the development, use, or evolution of free ERP and EB software modules. As such, it has no direct competitors in the traditional business sense of market share, sales and distribution channels, and revenue streams.

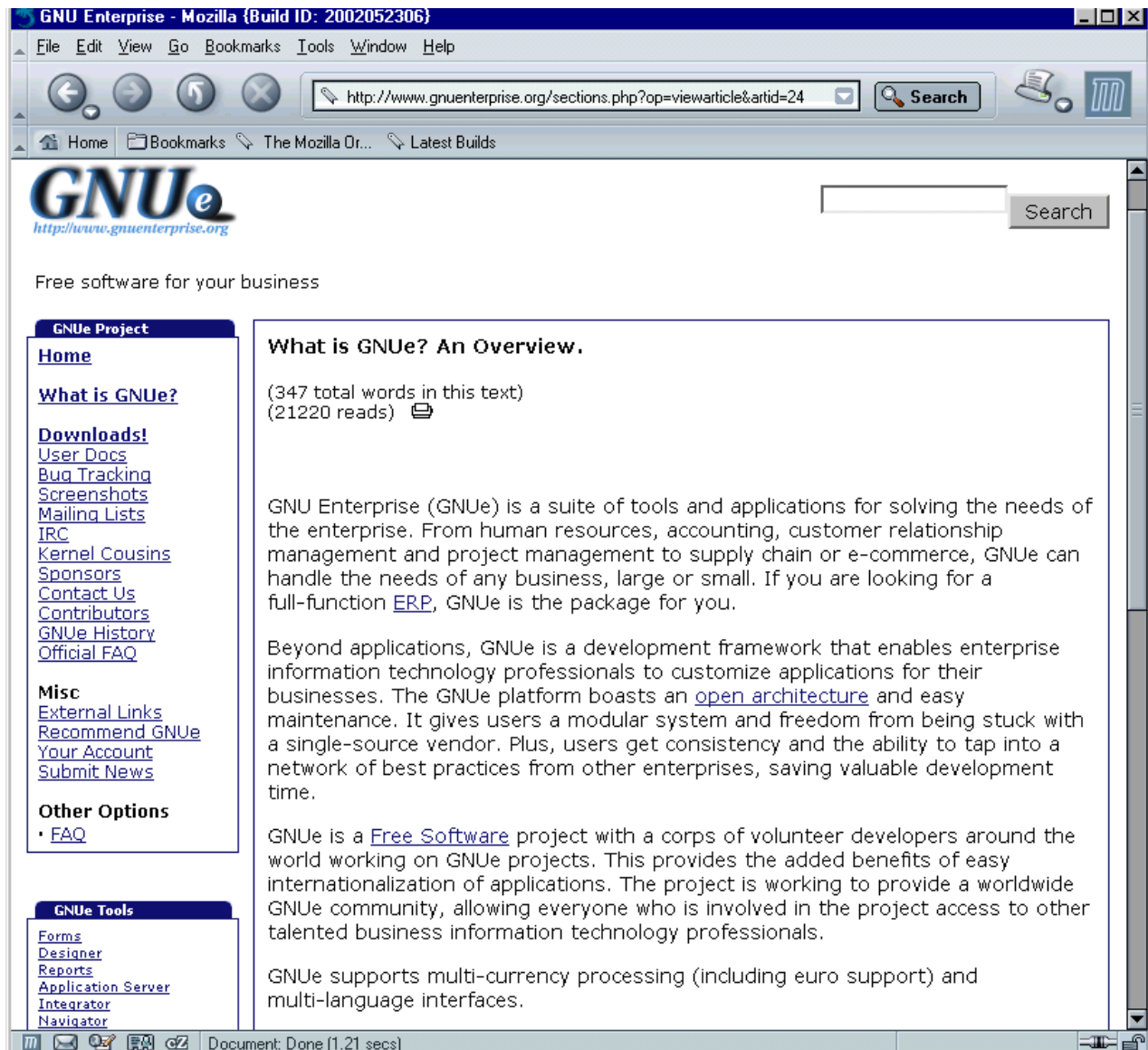


Figure 5. Display of background information on the GNUenterprise.org and its GNUe software (Source: <http://www.gnuenterprise.org/sections.php?op=viewarticle&artid=24>, June 2002)

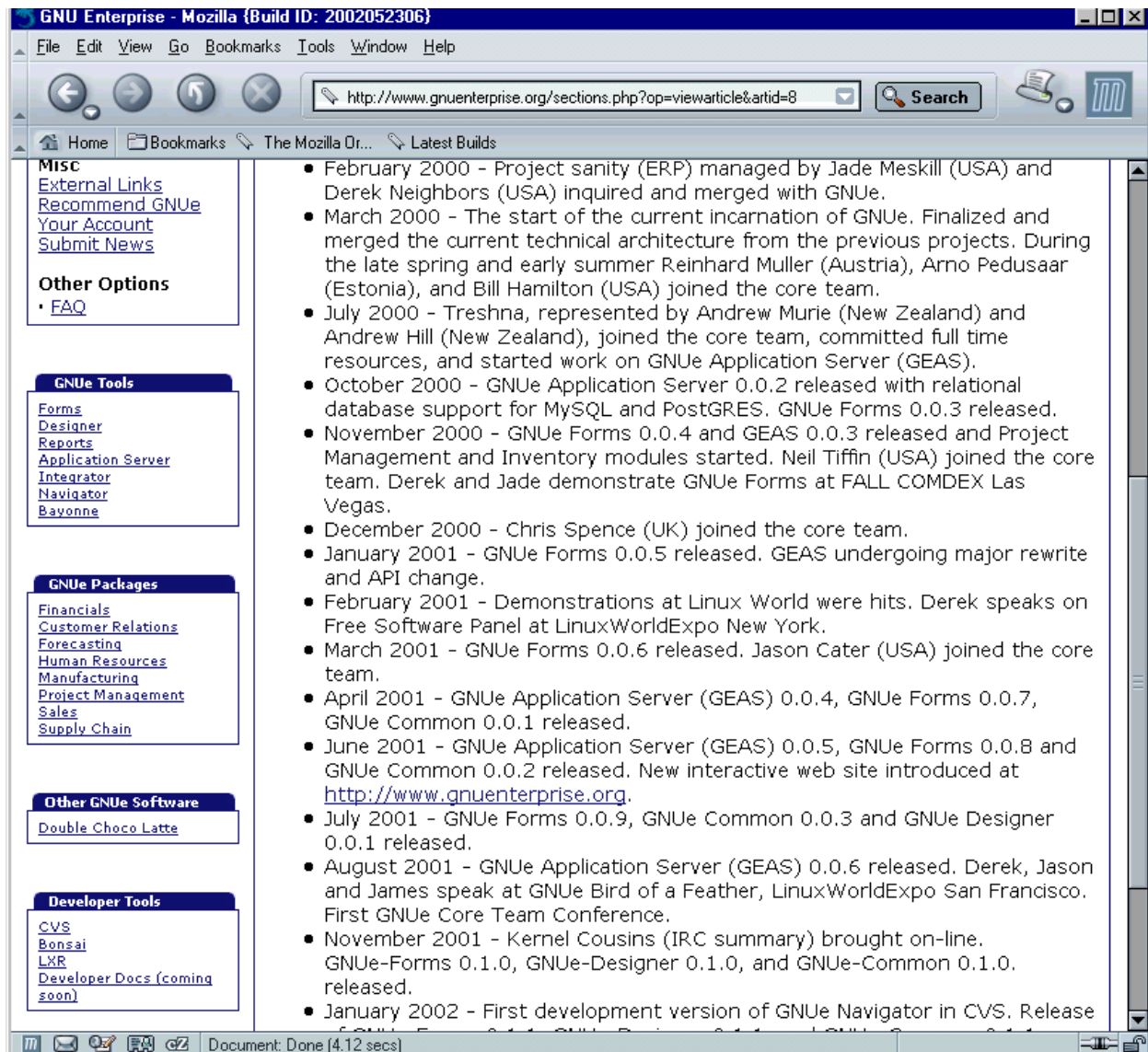


Figure 6. A view of the history of the development of GNUe software
 (Source: <http://www.gnuenterprise.org/sections.php?op=viewarticle&artid=8>, June 2002)

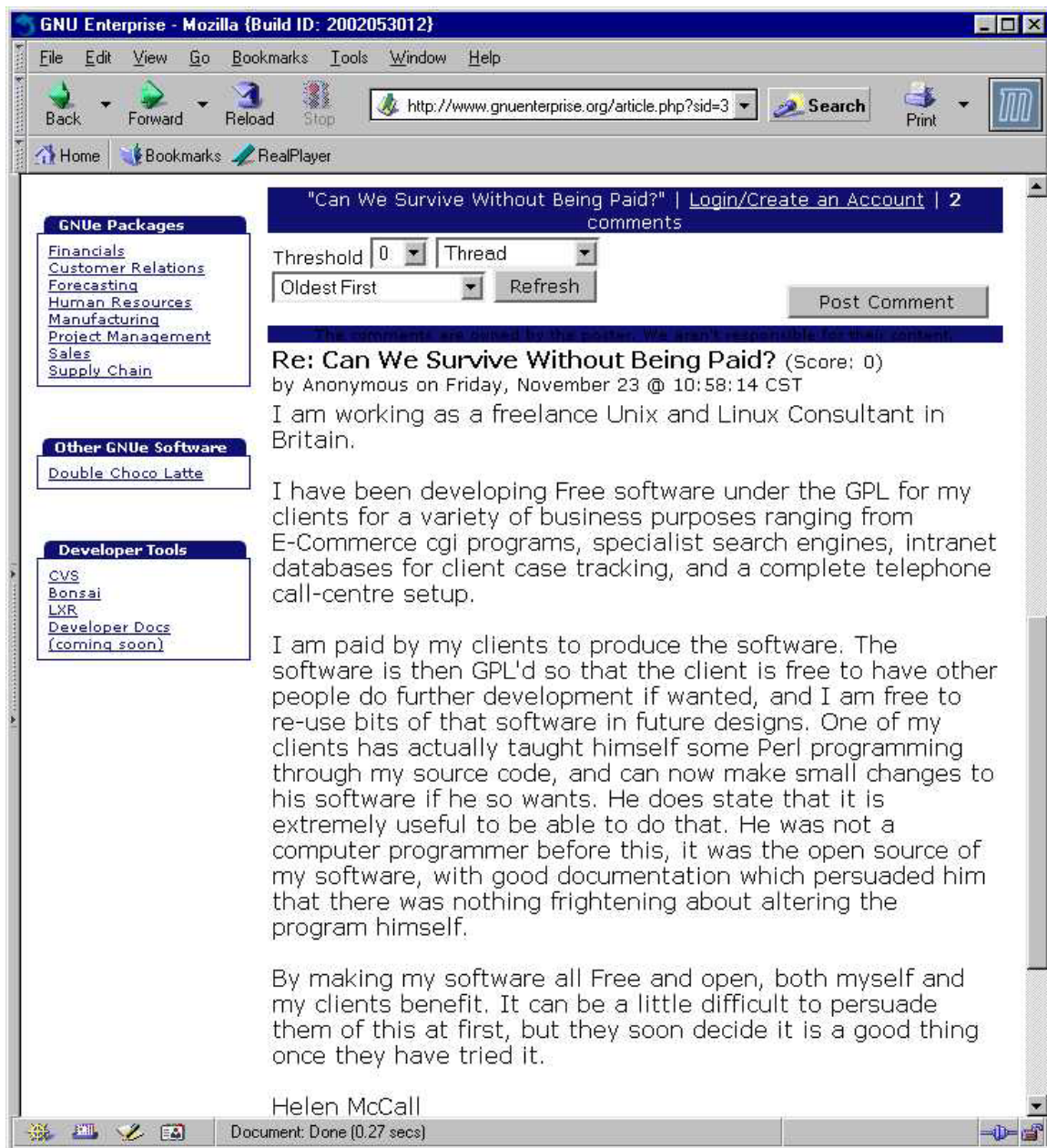


Figure 7. A public message from a GNUe project participant indicating her basis of financial support as a professional consultant and developer of Free software. (Source: <http://www.gnenterprise.org> June 2002)



Figure 8. A partial display of international companies providing resources and sponsoring the development of the GNUe software.

(Source: <http://www.gnuenterpise.org/sections.php?op=viewarticle&artid=6>, June 2002)

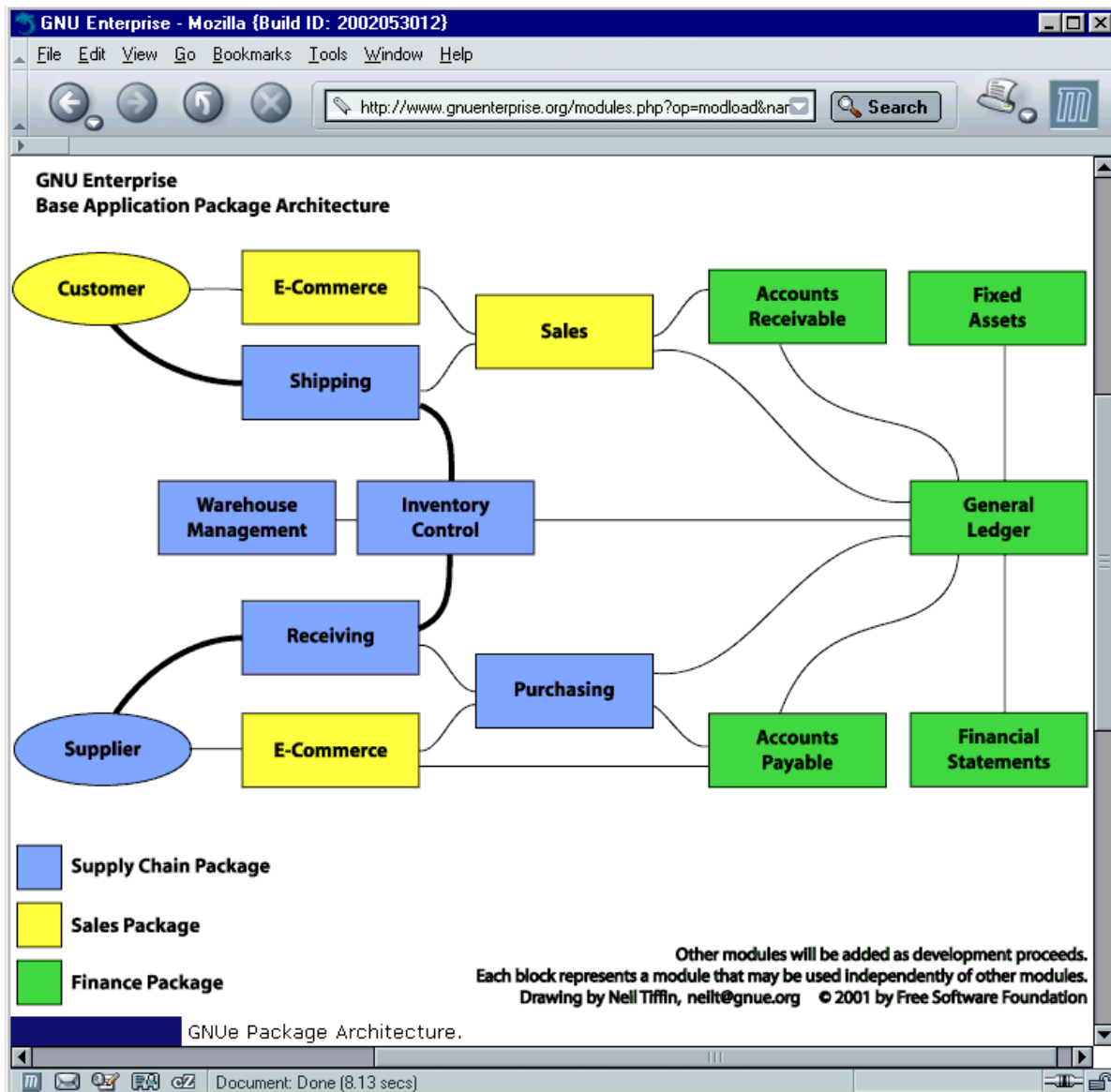


Figure 9. Overview of the GNUe software system architecture and modules
 (Source: http://www.gnuenterprise.org/modules.php?op=modload&name=NS-My_eGallery&file=index&do=showpic&pid=28 June 2002)

GNUenterprise.org does not represent a direct competitive threat to companies like SAP, Oracle, PeopleSoft, or JD Edwards. This will be true until these companies seek to offer low-cost, entry-level ERP or EB service applications for small business customers. However, it does compete for attention, participation, and mindshare from potential OSS developers/users with companies like Compiere and others that seek to develop and deploy OSS for ERP applications and EB service offerings. In addition, since the development of the GNUe software is open for global public review and corporate assessment, it is possible that the efforts and outcomes of GNUenterprise.org might influence other companies developing ERP or EB software. For example, other non-free, closed source ERP software vendors may perceive competitive pressure of new system features, lower cost software products, better quality, more rapid maintenance, or modular system architectures [CW360, 2002] arising from the globally visible OSSD efforts of GNUenterprise.org.

The GNUenterprise.org virtual organization is informal. There is no lead organization or prime contractor that has brought together the alliance as a network. It is more of an emergent organizational form where participants have in a sense discovered each other, and have brought together their individual competencies and contributions in a way whereby they can be integrated or made to interoperate [Crowston and Scozzi 2002]. In GNUenterprise.org, no company has administrative authority or resource control to determine: (a) what work will be done; (b) what the schedule will be; (c) who will be assigned to perform specified tasks; (d) whether available resources for the project are adequate, viable, or extraneous; nor (e) who will be fired or reassigned for inadequate job performance. As such, there is comparatively little administrative overhead to sustain ongoing going software development and community portal support activities. Instead, there is a group of core developers, secondary contributors, and casual volunteers who review and comment on what has been done. The participants come from different small companies or act as individuals that collectively act to move the GNUe software and the GNUenterprise.org community forward. Thus, the participants self-organize in a manner more like a meritocracy [Fielding 1999], rather than a well-orchestrated community for Web-based commerce or entertainment [Kim 2000].

Certain kinds of software development decisions are made by "logically centralized but physically distributed" core developers [cf. Noll and Scacchi 1999]. These core developers have earned the trust, have been recognized as technical authorities, and have achieved some degree of "geek fame", in the eyes other project participants [cf. Fielding 1999, Pavlicek 2000]. Like other project participants and OSS developers, the GNUenterprise core developers are expected to uphold and reiterate the freedom of expression, sharing, and learning that free, open source GNUe software represents or offers. So as core developers of GNUe software, they must reflect on how their software development decisions reflect, embody, or otherwise reproduce belief in free, open source software. On the other hand, decisions to contribute gifts of skill, time, effort, and other production resources that give rise to software, online communications, and technical peer reviews, are externalized or decentralized across a virtual organization [Bergquist and Ljungberg 2001, Crowston and Scozzi 2002]. These decentralized costs

reduce the apparent direct cost and administrative overhead of OSSD by externalization and global distribution, while sustaining something of a centralized decision-making authority. Thus, individual, corporate, and collective self-interest are motivated, sustained and renewed in a manner accountable to the culture and community that is GNUenterprise.org [cf. Monge *et al.*, 1998].

As such, these conditions make this study unique in comparison to previous case studies of EC or EB initiatives, which generally assume the presence of a centralized administrative authority and locus of resource control common in most large firms. Nonetheless, we still need a better understanding of what resources are brought to bear on the development and deployment of EB and ERP software by GNUenterprise.org. Subsequently, what follows is a description of key resources being employed throughout GNUenterprise.org to develop and support the evolution of the GNUe software modules.

Computing infrastructure resources

Developers contributing to the ongoing evolution of the GNUe software in general provide their own personal computing resources. This is especially true for unpaid volunteer contributors, but also true of salaried participants who are paid to work on the GNUe software, particularly for their work at home. There is no standard or common personal computer configuration that is defined as the development platform, other than those that can run either Microsoft Windows or GNU/Linux operating systems, and that they can access the Internet or Web as needed. Thus, all GNUenterprise.org community members must provide their own way into the project, via personal resource subsidies.

Beyond this, individual participants and contributors in the project are also expected to provide their own personal software development tools. These tools are generally expected to include those for source code development (e.g., code text editors, compiler collections, debugging tools, document formatters, local file repositories) and communications (Email clients, Web browsers). In addition, software contributors routinely use the shared project coordination tools such as the CVS software version repository manager [Fogel 1999], Internet Relay Chat (IRC) for instant messaging and message logging⁸, and emerging GNUe software modules. However, within the GNUenterprise.org community, there is a strong, often reiterated belief that project contributors should only use software tools that are also free, open source software or that support non-proprietary data exchange formats, rather than proprietary, closed source products or data formats.

There are shared computing resources that help support and embody the GNUenterprise.org effort. These include the Web servers that host the content, communications, and related software development artifacts associated with GNUenterprise.org community. The hardware side of the Web servers and their Internet service connection and fees are provided by companies that sponsors the GNUenterprise.org project. The project's Web servers include use of the Apache Web

⁸ Message logging provides a persistent record of IRC message streams, which then serve as a medium for capturing and documenting what is being discussed on a daily basis within the project. The GNUenterprise.org message logs are a rich yet public medium that serves as an online meeting room.

server and the PHP-Nuke content management system, which together provide Web portal capabilities while also serving as the community's information infrastructure [Deltor 2000].

Community communication resources

The GNUenterprise.org communication infrastructure serves as the “place” where GNUe software development work is performed, and where GNUe's software requirements are articulated, refined, stored, or discarded [Scacchi 2002b]. These communication systems, appear in the form of: (a) messages placed in a Web-based Threaded email discussion forums (Figure 10); (b) email list servers; (c) persistent, publicly available chat/IM records (Figure 11); (d) online community digests (Figure 12); and community news (Figure 13). Messages written and read through these systems, together with references or links to other messages or software webs, then provide some sense of context for how to understand messages, or where and how to act on them. Messaging also serves to enable informal communication about the availability of community participants with respect to different messaging forms, and their ability to coordinate and schedule their presence in online (virtual) project meetings [cf. Nardi, Whittaker and Bradner 2000].

Scenarios of usage resources

GNUe software developers generally do not meet face-to-face to create, employ, read, and revise shared mental constructions of how a given system is suppose to function. Since shared understanding must occur at a distance in space or time, then community participants create artifacts like screenshots, guided tours, or navigational click-through sequences (e.g., “Back” and “Next” Web page links). These scenarios of usage are augmented with supplementary narrative descriptions in attempting to convey their intent or understanding of how the system operates, or how it appears to a user when used. This seems to occur when GNUenterprise.org participants find it simpler or easier to explain what is suppose to happen or be observable at the user interface with pictures (or related hypermedia) than with just words. A later example in Figure 14 provides an example usage scenario that is intended for GNUe software application developers.

How-to guides, To-do lists, FAQs, and other itemized list resources

Online documents that capture and condense “how to” perform some behavior, operation, or function with the GNUe system, serve as a semi-structured narrative that assert or imply end-user requirements [Scacchi 2002b]. Formal HowTo descriptions include explicit declarations of their purpose as a HowTo, and thus may be identified as a system tutorial. Community participants may seek these formal HowTo's when they are unfamiliar with how to add a system module or class structure, or contribute other resources or efforts to the open software project. In contrast, informal HowTo's may appear as a selection, composition, or recomposition of any of the proceeding resources. These informal HowTo guides may be labeled as a “FAQ”, such as that shown in Figure 15. A FAQ is a list of frequently asked questions about how a system operates, how to use it, where to find it's development status, who developed what, known bugs and workarounds, etc. However, most FAQs do not indicate how frequently any of the questions may have been asked, or if effort has been made to measure or track FAQ usage/reference.

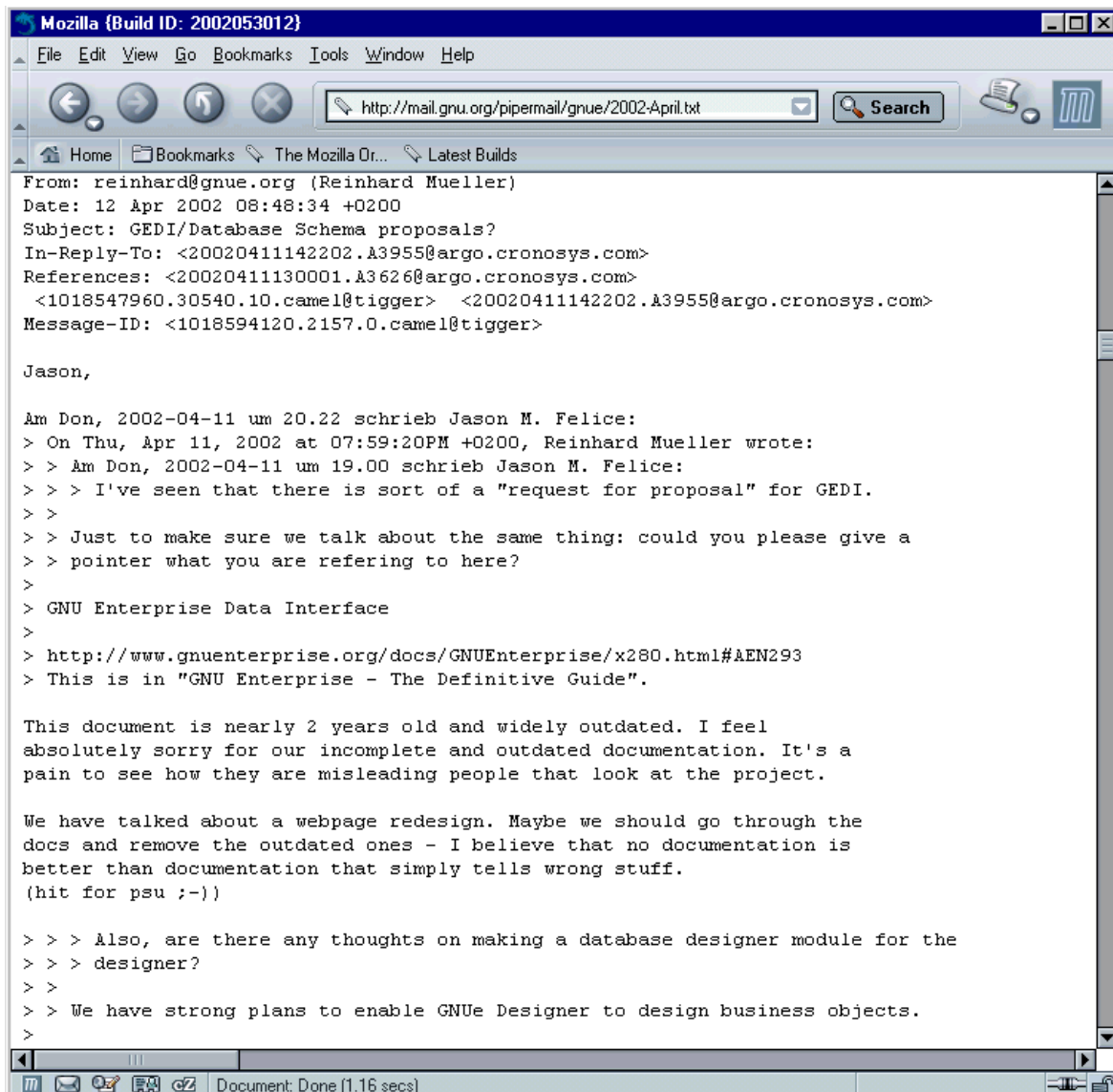


Figure 10. A view of a public email thread transcript communicating project information and evaluative statements between project participants
(Source: <http://mail.gnu.org/pipermail/gnue/2002-April.txt> April 2002)

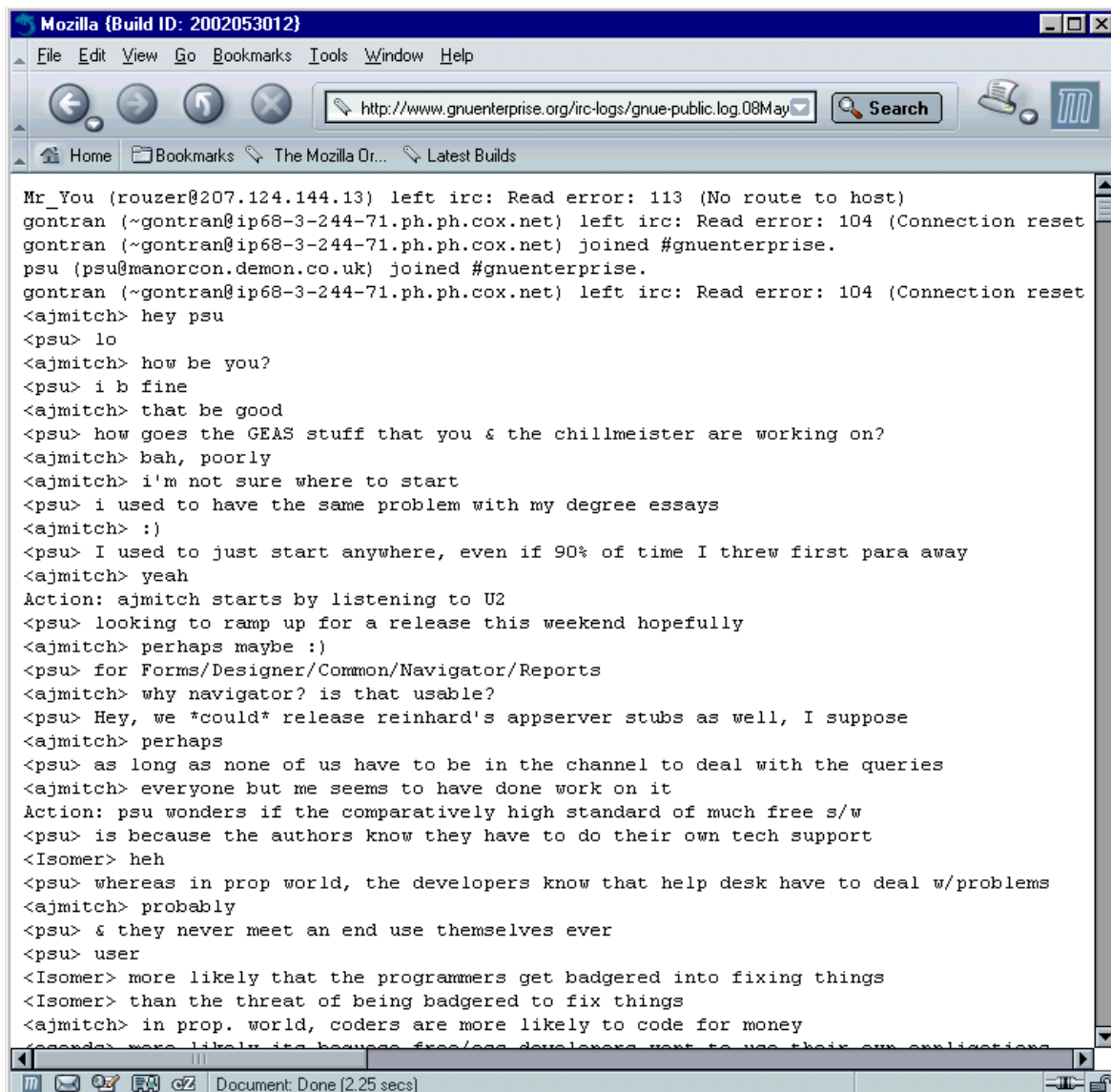


Figure 11. A public transcript that records Internet-based chat in a real-time online project meeting

(Source: <http://www.gnueenterprise.org/irc-logs/gnue-public.log.08May2002> May 2002)

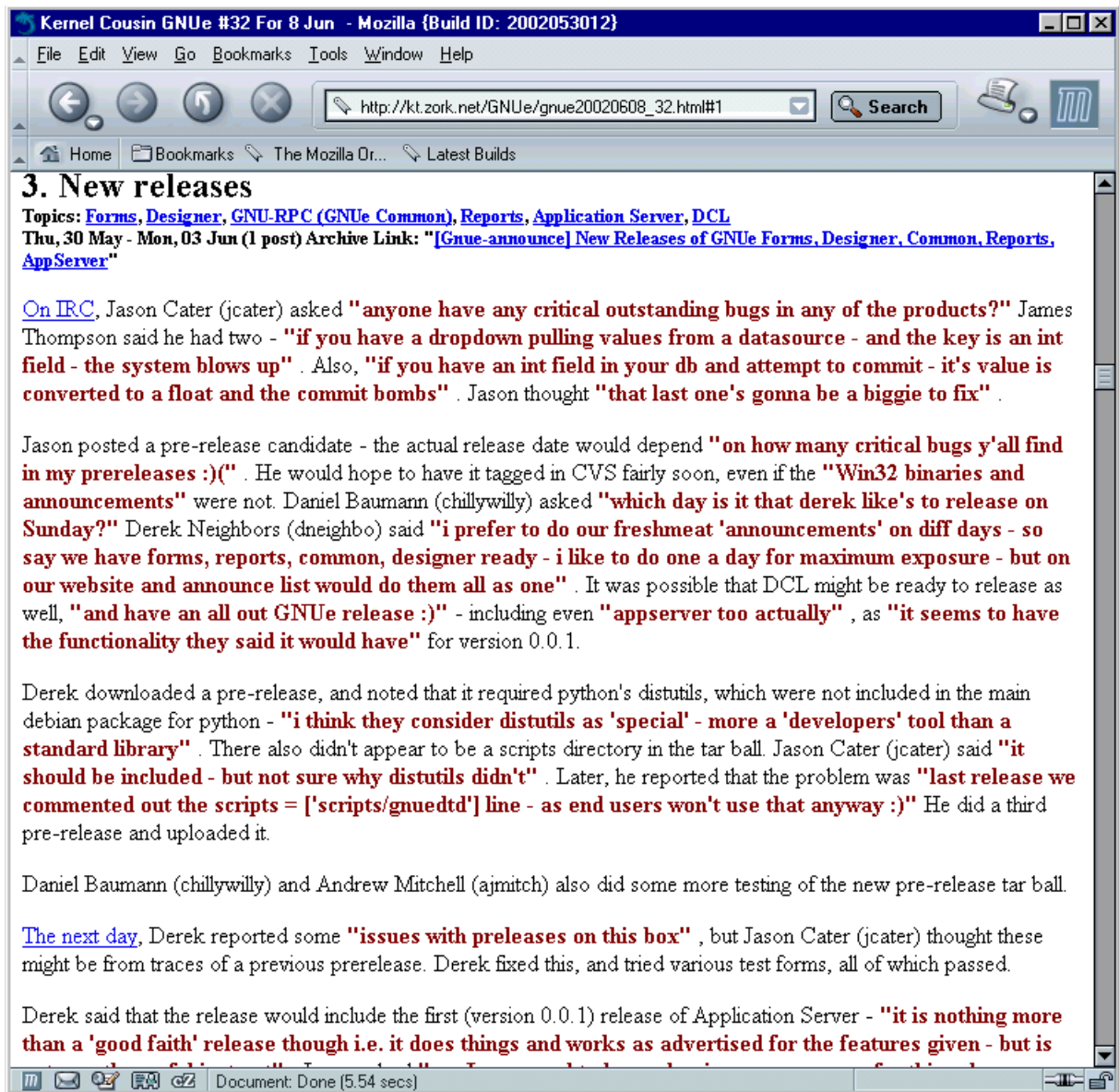


Figure 12. A snapshot of a community digest summarizes communications and activities across the GNUe project, with embedded Web links to source documents
(Source: http://kt.zork.net/GNue/gnue20020608_32.html#1 June 2002)

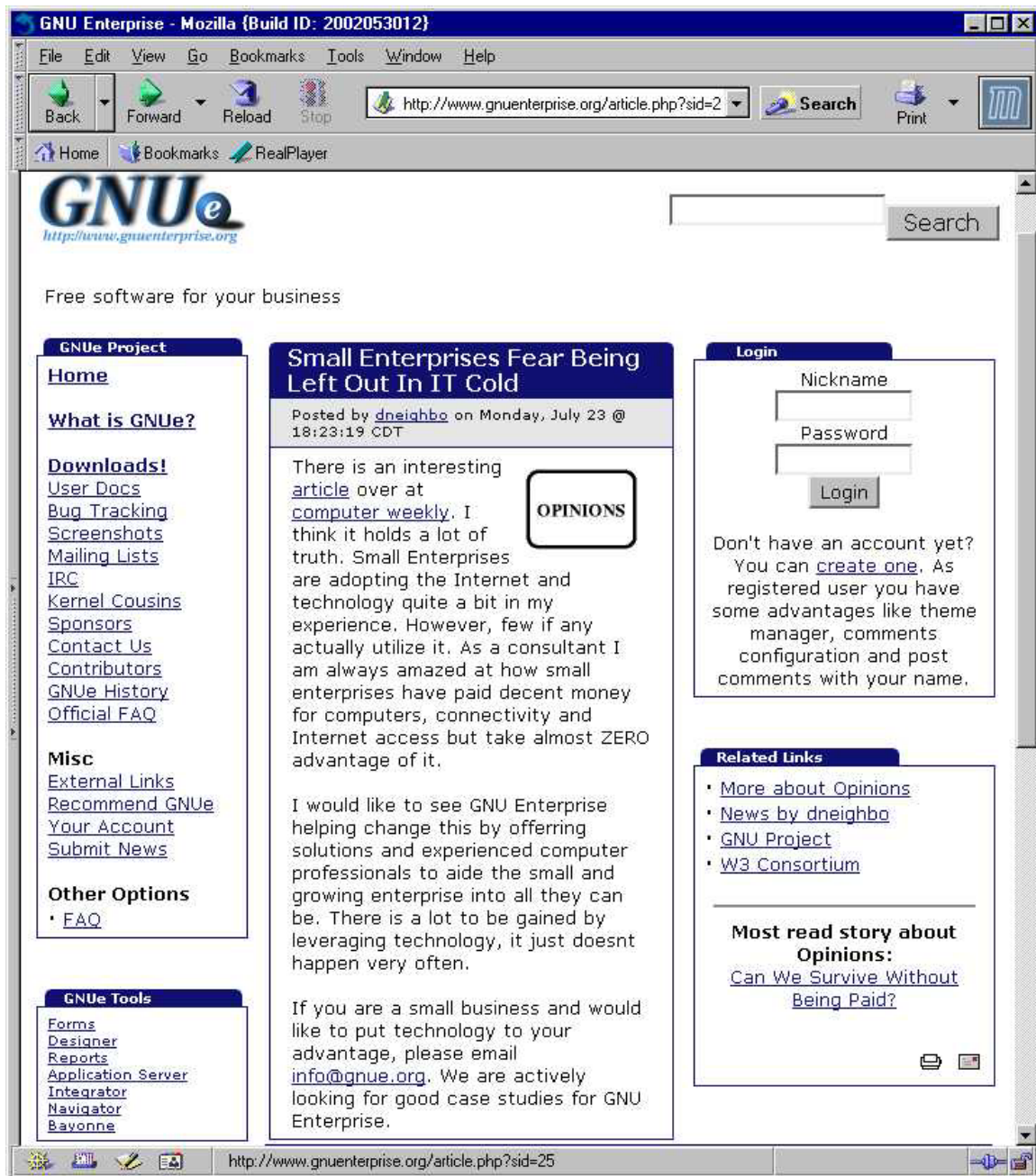


Figure 13. A display view of a news item of interest to GNUenterprise.org participants
(Source: <http://www.gnuenterprise.org/article.php?sid=25>, July 2001)

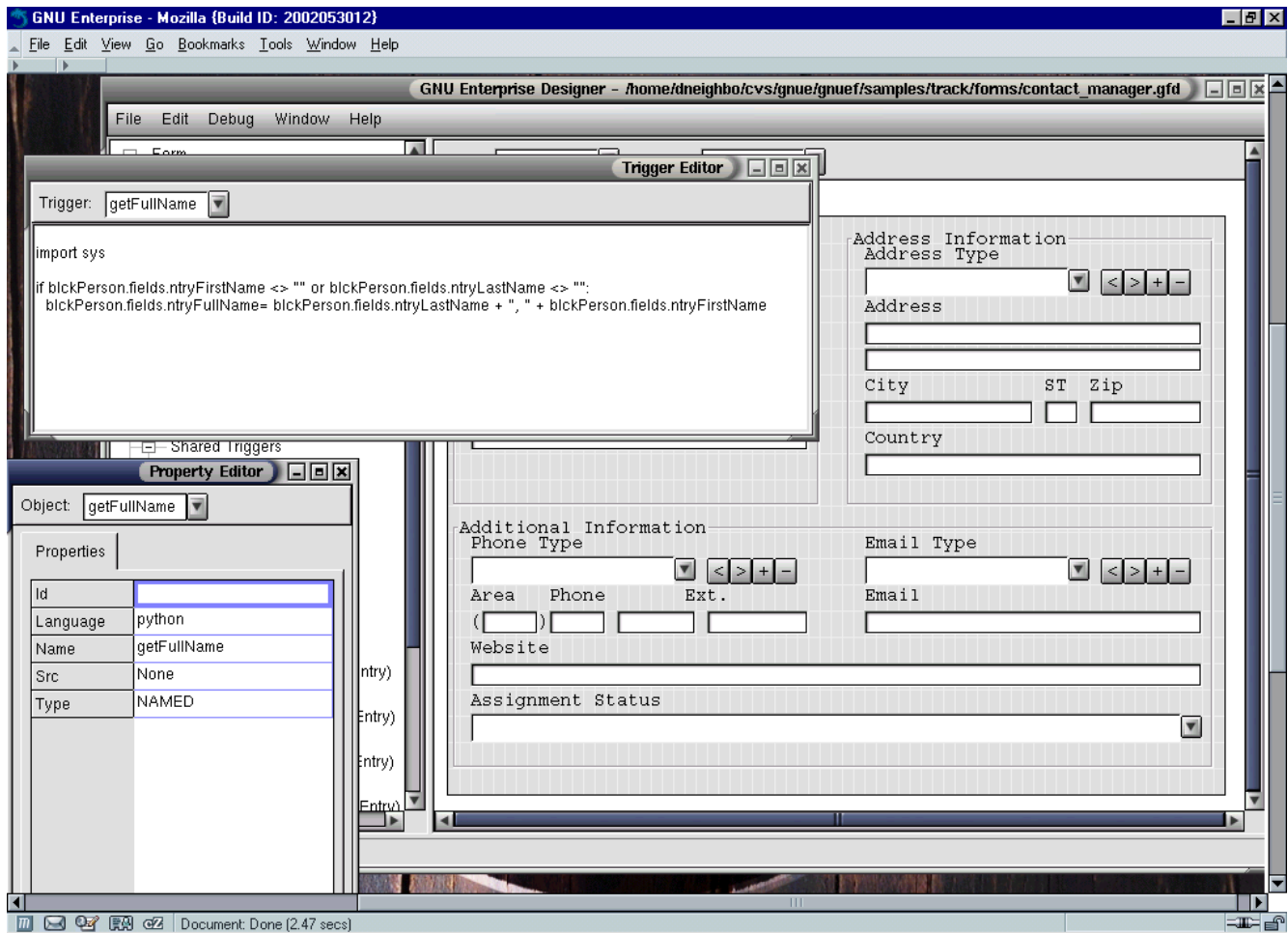


Figure 14. The GNUe Enterprise Designer, an interactive development environment (IDE) displaying a usage scenario for developers of GNUe-based applications (Source: www.gnueenterprise.org screenshot collection, May 2002)

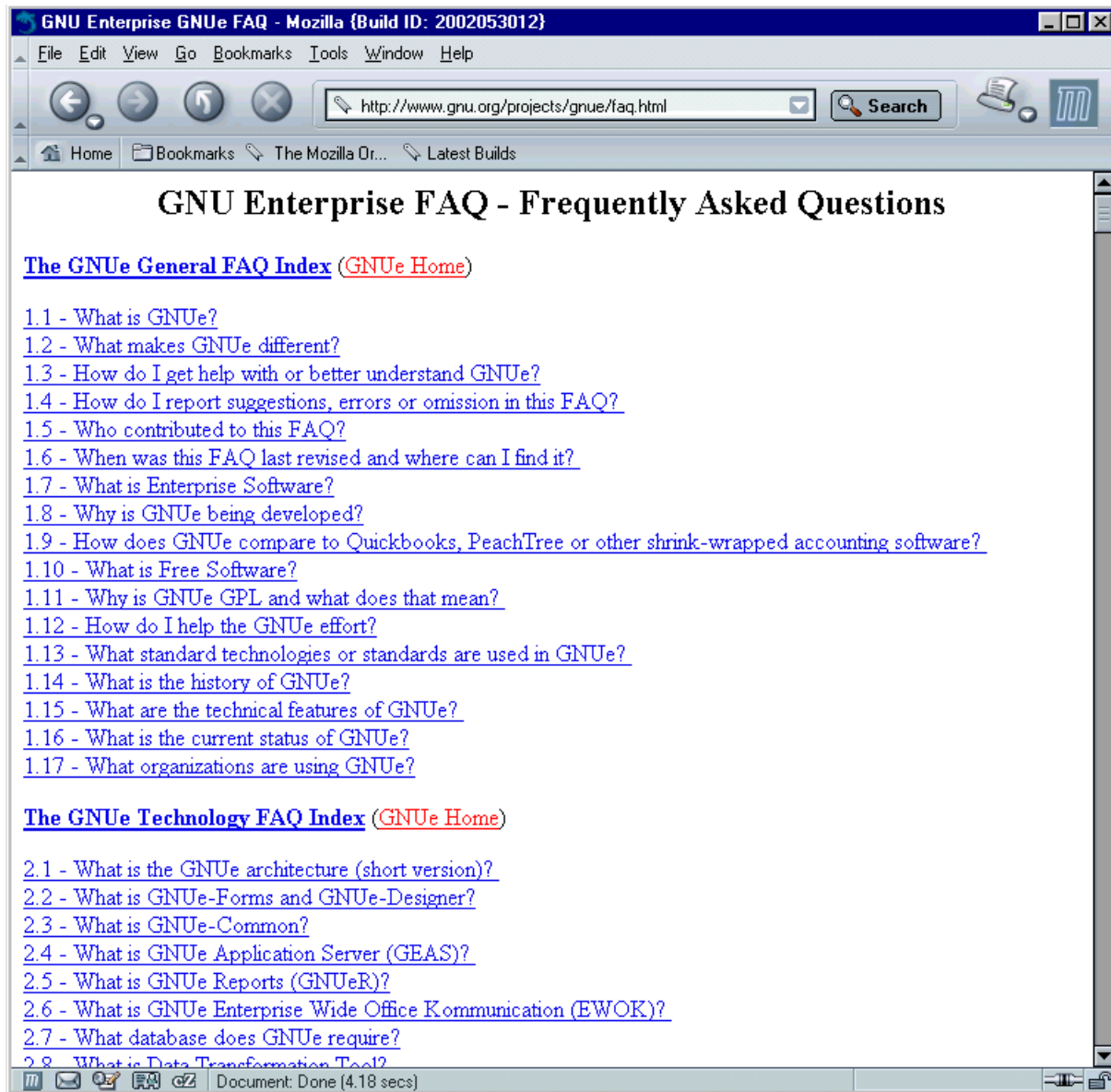


Figure 15. A screen display of links to an itemized list of "Frequently Asked Questions"

(Source: <http://www.gnu.org/projects/gnue/faq.html> April 2002)

Traditional system documentation resources

GNUe software modules have online system documentation or documentation suitable for printing in support of end-users or developers. It was possible to locate online user manuals or help pages that describe commands and command parameters for how to (further) develop, invoke or use GNUe system modules (Figure 16). Whether the online documentation was up-to-date or dated, and subsequently (in)consistent with current functional capabilities or system commands was unclear. But this is no different from conventional software systems whether developed in-house, or acquired in closed source form via a commercial product.

External publication resources

A small but growing number of OSSD projects have become the subject of external research or trade publications available in global audiences [Scacchi 2002b]. GNUenterprise.org has not yet realized any specific publications to draw attention to its efforts. However, it seems likely that some participants of the GNUenterprise.org are likely to have encountered or been influenced by publications on Free Software [Williams 2002] to help galvanize their interest in participating and contribution to the GNUenterprise.org software development community. More importantly, as a growing number of national governments outside the U.S. begin to move towards public policy statements that encourage or mandate the use of free and OSS for E-Government applications of ERP and EB applications [cf. Cataliana 2002, Nunez 2002, Scacchi 2002d], it seems reasonable to expect more attention and potential investment will appear from outside the U.S. for projects like GNUenterprise.org. Thus, external publications that bring increased public attention and awareness to the GNUenterprise.org project will enhance its prospects for community growth and for increased software functionality and interoperability.

Open Web site and software resources

Web site and Web portals are central to the ongoing existence, public awareness, and development of a community of interested people. GNUenterprise.org is no exception, and in fact relies on different kinds of Web portals to link itself to the broader community of people interested in free and OSS projects. First among these is the project Web site itself, found at www.gnuenterprise.org. This site serves as a portal to everything associated with the ongoing development of the GNUe software, as well as the community of people who participate [cf. Deltor 2000]. For example, Figure 17 displays a view into GNUenterprise.org where one can find the file directories that contain the GNUe software in source code form. Elsewhere, the GNUenterprise.org Web site is in turn dependent on three closely related sites that indirectly support the GNUenterprise.org effort, as well as help drive prospective contributors or end-users (i.e., virtual project staff) to the site. This includes the portal at SourceForge.net, Savannah.org, and PHP-Nuke.org. Beyond these are the many individual and corporate Web sites that have established and highlight Web hyperlinks that reflect their interest or association with the GNUenterprise.org effort.

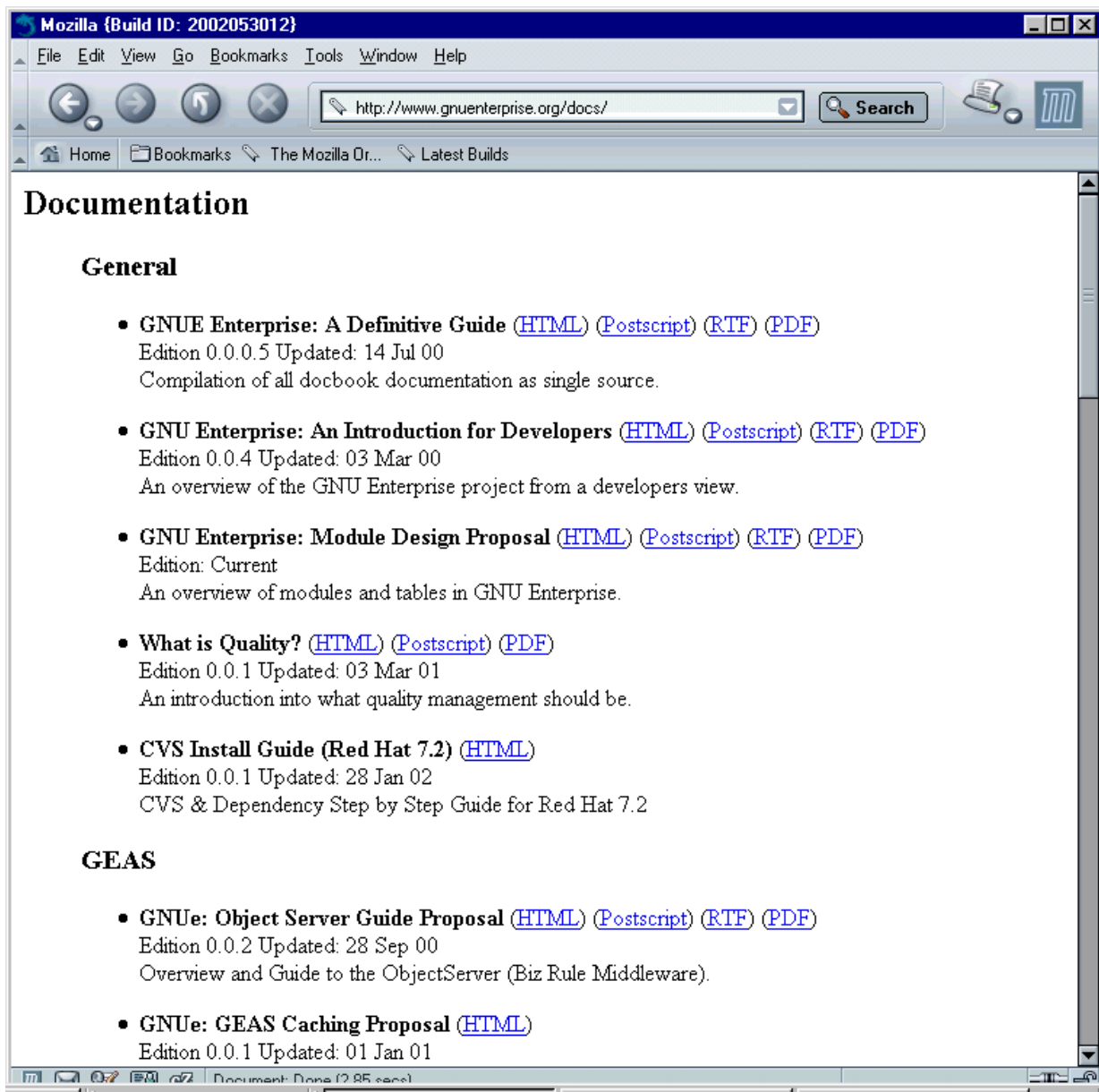


Figure 16. A view of a list of online user documentation found at GNUenterprise.org
(Source: <http://www.gnuenterprise.org/docs/> June 2002).

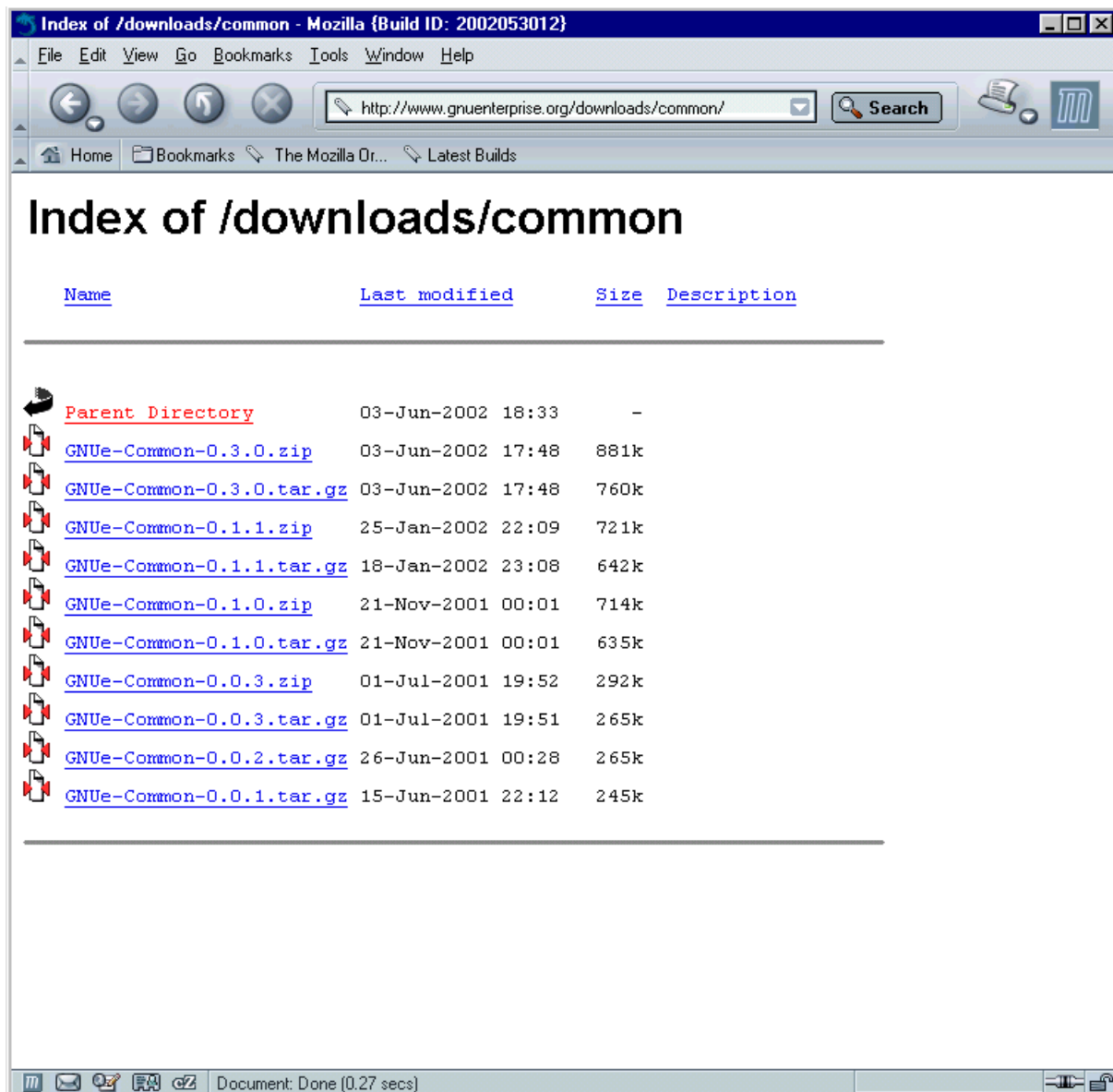


Figure 17. A display of a view into a source code Web directory associated with a GNUe software distribution version.

(Source: <http://www.gnuenterprise.org/downloads/common/> June 2002).

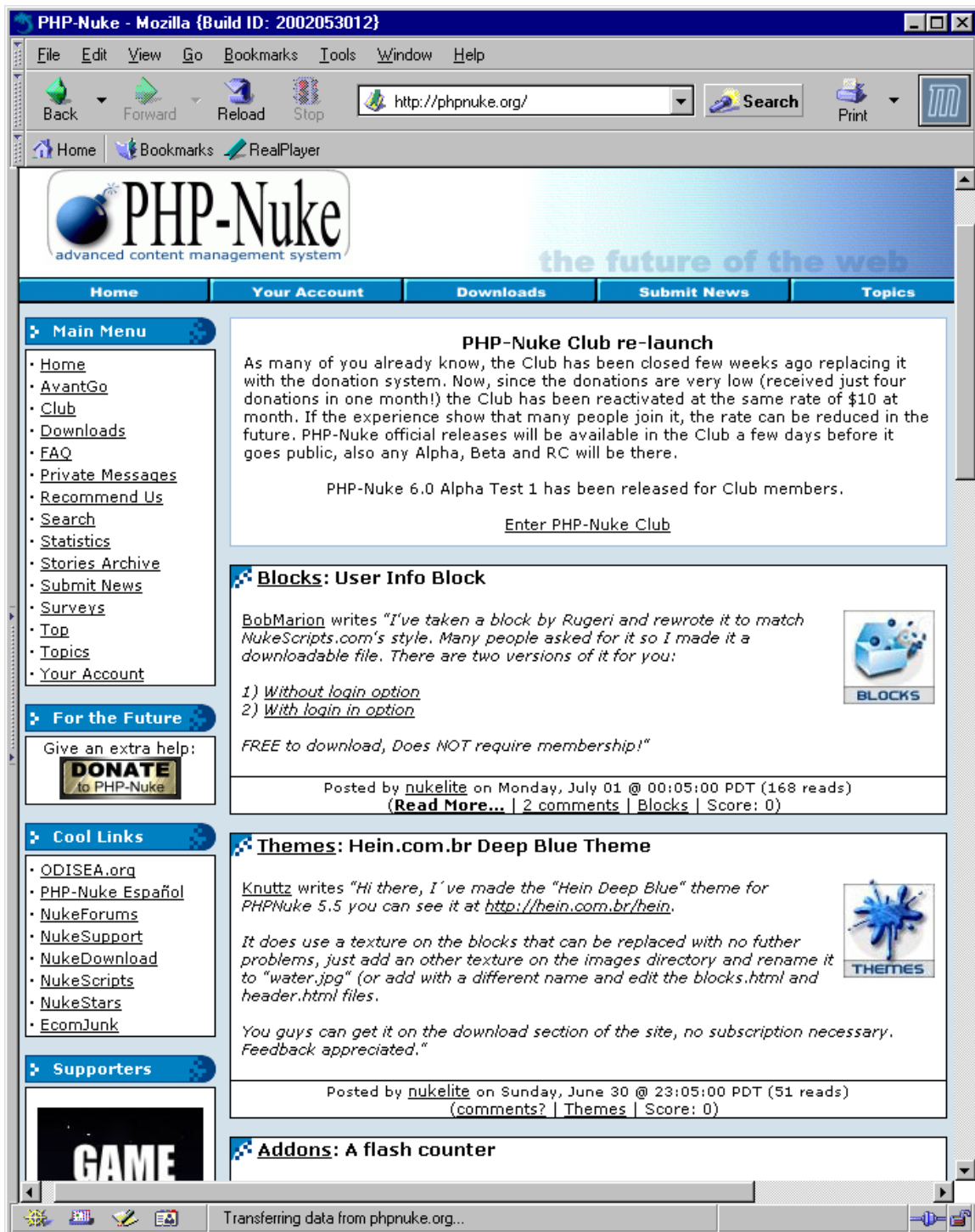


Figure 18. A display of the Web site for acquiring and discussing a free, open source software content management system that supports the GNUenterprise.org Web site. (source: <http://phpnuke.org/> June 2002).

SourceForge.net is the world's largest directory of OSSD projects as noted earlier. GNUenterprise.org has a presence at SourceForge, as shown in Figure 18. However, the Compiere OSS ERP project has a much larger presence at SourceForge, recently appearing as a "top 10 most active" site for a number of weeks in 2002, as indicated in Figure 18. GNUenterprise.org also has a presence at Savannah.org, which is the portal site for all free software development projects affiliated with the Free Software Foundation.

In contrast to SourceForge.net, Savannah.org hosts about 1000 projects, but all of the projects are expected to adhere to the principles, guidelines, and license terms associated with free software [cf. DiBona, Ockman, Stone, 1999, Pavlicek 2000, Williams 2002]. Thus, Savannah.org may help drive potential contributors and end-users to GNUenterprise.org because it is developing an important class of application software (ERP, EB), and because it is a free software development project, not just an OSSD project.

Finally, PHP-Nuke.org (found at www.phpnuke.org) is a free software project focussing on the ongoing development of content management systems that serve as the software base for complex Web portals like GNUenterprise.org. PHP-Nuke.org is not likely to drive traffic to GNUenterprise.org⁹. Instead, GNUenterprise.org relies on the availability of the PHP-Nuke CMS software to help sustain and serve its presence on the Web, as well as to potentially offer the ability to integrate free CMS software into the ERP and EB modules being developed by GNUenterprise.org.

Software extension and customization resources

The GNUe software is not a monolithic, tightly integrated software package. Instead it is a collection of software modules or subsystems that can be configured and extended in a variety of ways that conform to an open modular architecture (cf. Figure 9). To support this architectural flexibility in ERP or EB application design, the GNUenterprise.org development team provide and employ an interactive development environment (IDE) that is specialized for GNUe software development. Furthermore, some of the GNUe software modules provide generic functionality that can be used by other GNUe modules. An example of this is the GNUe Forms module that is used in the development of graphic user interface forms for data entry. Both the IDE and the tailorable EB Forms module are shown in the scenario displayed in Figure 14. Accordingly, the IDE and GNU Forms module are reusable and embeddable resources that enable the extension and customization of the other GNUe modules.

⁹ A search at www.phpnuke.org for GNUe, gnue, GNUenterprise, or gnuenterprise produce no matches, thus suggesting there is probably no Web hyperlink from PHP-Nuke.org to GNUenterprise.org

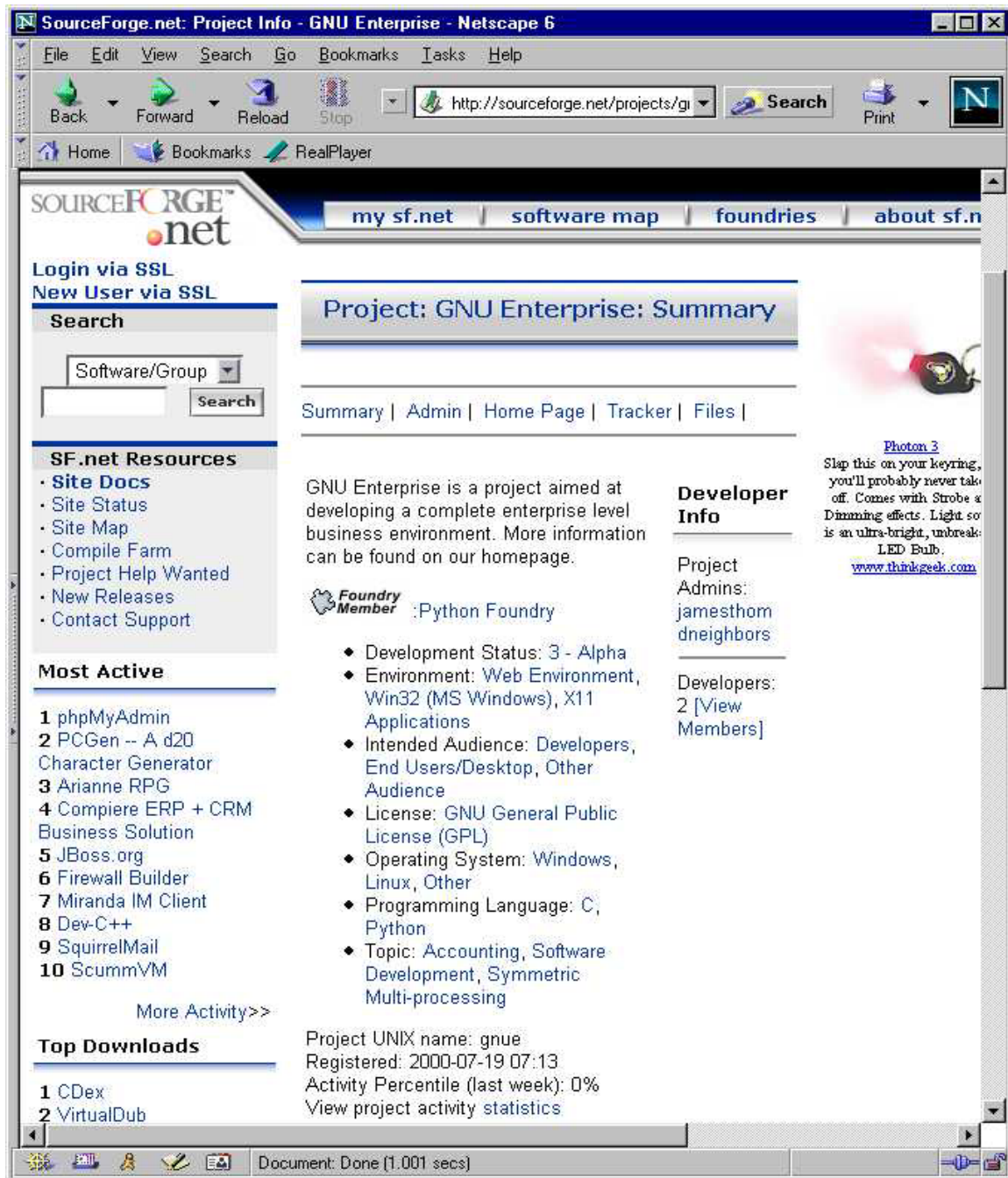


Figure 19. Display showing a view of the SourceForge entry for GNUenterprise.org
(Source: <http://www.sourceforge.net/gnuue> June 2002)

Elsewhere in the GNUenterprise.org project, a closer look at some of the ERP and EB system modules being developed as part of the GNUe software reveals a degree of reliance or reuse of software from other free software projects. These include the GNU Image Manipulation Package (GIMP) and GNU Bayonne, an enterprise-scale telecommunications application server. GNUenterprise.org has also recently merged with the Double Choco Latte (DCL.sourceforge.net) free software project and development community. DCL is a basic project management, work order, and help desk management system. GNUenterprise.org relies on DCL to manage and keep track of open issues and bugs found in the GNUe software (Figure 20). Thus, this ability to use, reuse, or otherwise leverage software from other free software development projects and Web sites, provides a powerful set of software development resources and infrastructure that are often beyond the reach of companies that limit themselves to proprietary, closed source software development.

Free software license resources

The GNUe software is being developed and distributed under the auspices of the GNU Public License. The GPL is the premier copyright license for promoting and sustaining free, open source software. The GPL reiterates the rights and freedoms identified earlier and thus bestow them onto GNUe software modules. Figure 21 provides a view of the GPL that is associated with the GNUenterprise.org project, which in fact is the GPL maintained by the Free Software Foundation.

A frequently noted concern regarding software protected under the GPL is that its property rights may be "viral". This suggests any proprietary software that is used in conjunction with software like GNUe must therefore also become subject to the freedoms and rights assigned by the GPL. Though there has been no legal test or substantiation of this, it does not appear that merely using free, GPL protected software like GNUe, together with non-free proprietary software like an Oracle database management system or Microsoft Windows operating systems, makes either the Oracle or Microsoft software suddenly "free". Similarly, no one seems confused about SAP offering a proprietary closed source ERP system that can interoperate with either a GPL protected database management system like SAP DB, or with a proprietary DBMS from Oracle or Microsoft.

The GPL as it applies to the GNUe and other free, open source software declares that any subsequent software application that incorporates, integrates, or internalizes the GNUe software, the resulting encapsulating software application would become subject to the GPL. For instance, if SAP decides to include some or all of the GNUe software within one of its products, then that software product (and not all SAP products) is subject to the property rights provided by the GPL. Of course, there is no apparent force that would require SAP to pursue such a course of action. Thus, the concern over when GPL protected software like GNUe is viral, and is somehow capable of infecting proprietary software with GPL freedoms, is erroneous or misleading. Instead, what the GPL provides and insures to the GNUe software modules is that they will remain free into perpetuity, and serve as a reusable or renewable resource for software development, use, modification, learning, and redistribution.

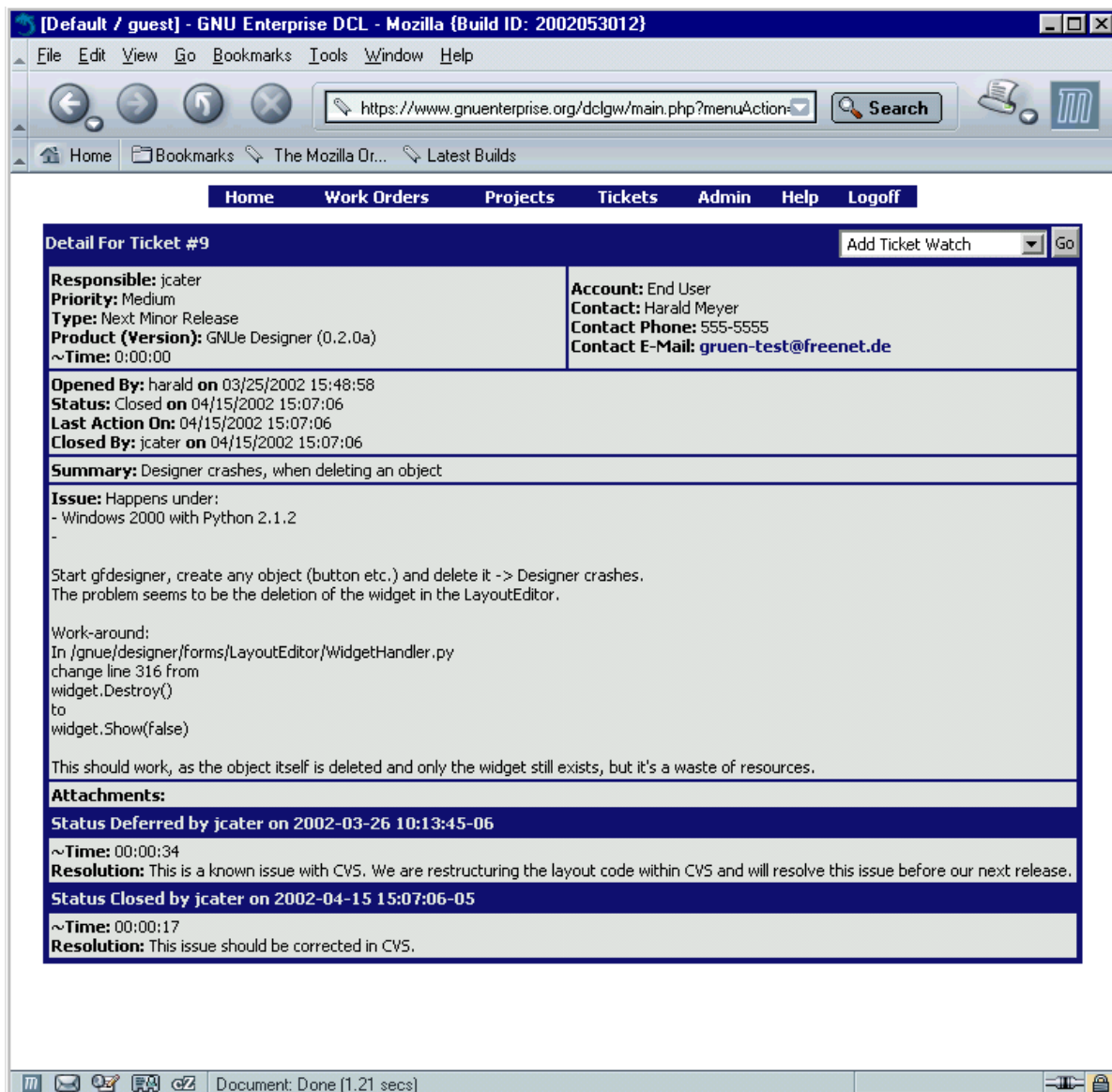


Figure 20. A software bug report and status information maintained in DCL that is associated with a GNUe software component (Source: <https://www.gnueenterprise.org/dclgw/main.php?menuAction=boTickets.view&ticketid=9> May 2002)

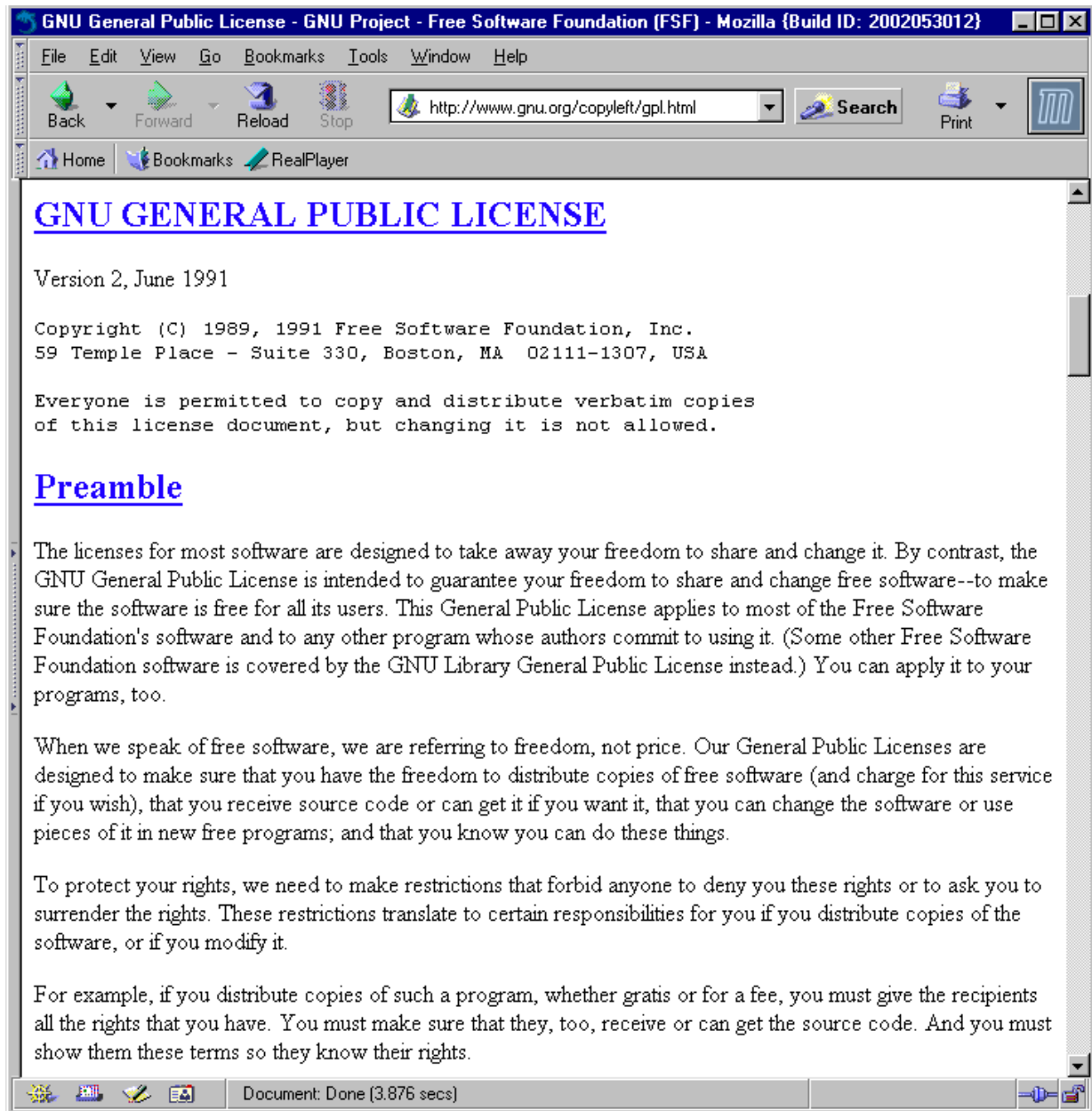


Figure 21. The community property and copyright license that covers all GNUe software
(Source: <http://www.gnu.org/copyleft/gpl.html> as linked to GNUenterprise.org, June 2002)

GNUenterprise.org software development capabilities

Given the virtual organization setting and resources identified for the GNUenterprise.org project, we now turn to examine how participants bring these together through individual and collective effort to develop the GNUe software. Perhaps, it should not be surprising that the development process in the project is not like those typical to centrally managed, staffed, and budgeted "software development life cycle" projects found in traditional corporate settings [Scacchi 2002]. Instead, the development process at GNUenterprise.org embodies a different kind of resource-based product development capability.

The overall development of the GNUe ERP and EB software modules consists of two interrelated and iterative processes. One is the free OSS development process, and the other is the GNUenterprise.org community development process.

The GNUenterprise.org software development process

The GNUe software development process entails an ongoing evolutionary cycle of activities performed by its core developers, contributing developers, volunteer reviewers, and early end-users. The following kinds of incremental activities can be observed, though they do not necessarily appear in the order presented:

Downloading, installing, and using the current versions of the GNUe software modules. You cannot begin to understand or contribute to the GNUenterprise.org project until you have a copy of the software that you can try out or otherwise review and evaluate. Furthermore, as the current version is continuously subject to changes and improvement over a period of days to weeks, then this becomes a routine activity for project participants.

Reviewing, analyzing, or redesigning existing project software, Web site content, or other related artifacts. Once a prospective contributor has acquired some familiarity and experience with the current software version, they may consider if there is some flaw that could be eliminated, or some improvement in software functionality that could be made. This is because in a free, OSSD project, the source code and other project related software/information resources, are treated by participants as community property, not private property. Thus, as a reviewer or contributor it is not a participant's job to criticize the source code or documents of an individual. Instead, criticism must be directed towards the community in order to help engage or motivate someone within the community to come forward and affect a change. Oftentimes, it is easiest for the person who discovers the flaw or asserts the improvement option to simply take responsibility or "ownership" for affecting the change. However, the person must still convince others in the project who care about the efficacy of such action. Successfully convincing others is not guaranteed, and thus may require sustained effort, revision of the proposed change, and multiple approve, reject, or rework votes [cf. Fielding 1999].

Communicating experience with current software versions. Much of the work in a free, OSSD centers around communication. Since there is no explicit project management

regime in effect, then project participants have become comfortable discussing how to improve the quality of the community's contributions. Anyone can potentially pose or answer a question, or offer a comment on the public activities of other project contributors. Contributors most knowledgeable about a question may refer the questioner to some existing project information resource, if a suitable answer can be found there. These referrals help less familiar contributors to learn more about how things work in the project, as well as who knows what, where, and when about different project resources or activities [cf. Ackerman and Halverson 2000].

Both synchronous IM tools like IRC, and asynchronous Email and threaded Email discussion groups (i.e., project bulletin boards) are routinely employed. In addition, the GNUenterprise.org project has adopted the ongoing practice of producing weekly community digests (called "Kernel Cousins") to provide an overview of recent events and discussions, which in turn help create a context for project participants who may not have been active in the past week or beyond.

Determining software requirements and design. The GNUe software is not developed in a manner that is suggested by common Software Engineering textbooks. There is no "software requirements specification" document, nor is there a formal software requirements elicitation, capture, specification, and document the sub-process [cf. Scacchi 2002b]. Instead, the requirements and design of the GNUe software modules is better seen as being continually emerging, though not in constant flux. The core developers play a central role in determining and discussing what software functionality is needed, and how it is to be configured or organized (i.e., designed) in order to maintain an open, modular system architecture. These determinations and discussions are conducted via IRC sessions and threaded Email transcripts, as shown earlier in Figures 10 and 11. The outcomes are posted on the GNUenterprise.org Web site as public message content, or as some related software informalism [cf. Scacchi 2002b].

Developing free open source code. This is where the programming work occurs. Core developers often take ownership of some portion of the emerging GNUe system or specific software modules. Ownership implies responsibility to the community, rather than assertion of personal property boundaries. Subsequently, software owners may take on more management-like activities, rather than programming activities.

Programming free OSS is an activity that mostly happens outside of the GNUenterprise.org workplace, but within the local personal computing workspace of individual project contributors. Developers may communicate snippets of code they are currently working on or are examining with other contributors. However, this is likely to occur while a developer is using a personal computer workstation with multiple windows (or "sessions") open on his/her screen, with one or more running local software development tools, and one or more others connected to the GNUenterprise.org Web site or messaging systems. Competent developers can then cut and paste information from one session to another, while working on multiple project-related tasks ("multi-tasking").

Managing and coordinating software development resources and developers. As many free, OSSD projects rely on the shared leadership of core developers [Fielding 1999], then there needs to be some way for the efforts and contributions of mostly autonomous developers to be brought together to "make" a complete current version of the overall software system or software module subsets. This entails using a software configuration and coordination tool like CVS [Fogel 1999] to organize and track which files constitute which modules in what version of the system. As use of CVS is pervasive throughout most OSSD projects, developers who access a CVS repository have a common shared view of current system versions and the process by which it is updated [Herbsleb and Grinter 1999].

A core developer or some other person so designated uses a centrally administered CVS repository to browse, build, regression test, and archive the current software version. The current version may exist in multiple variants, one for each of the targeted operating systems it runs on. Each variant must also be archived. Once archived, an executable (build) images of the software can be packaged for distribution and deployment from the GNUenterprise.org Web site. News about the existence of a new archived version can then be posted on the Web site or distributed via Email or newsgroup announcement, so that interested parties might download them and repeat the overall software development process cycle described above.

The GNUenterprise.org community development process

The GNUenterprise.org development process entails an ongoing cycle of incremental activities performed by its core developers, contributing developers, volunteer reviewers, and early end-users. The following kinds of activities can be observed, though they do not necessarily appear in the order presented:

Growing a community to develop free, open source software. Free OSSD projects appear more likely to get off the ground if there is some initial seed of free software to grow a project. GNUenterprise.org had such a free source code seed dating back before 2000. With the existence of seed code, and the discovery of like-minded software developers (see details within Figure 6), it becomes possible to establish a free, OSSD project. The announcement of such an event may coincide with the creation and posting of the project's Web site/portal. With a Web presence at hand, it then becomes possible to post project content like an itemized "to do" list, as a way to invite would-be contributors to join in by seeing if what needs to be done matches their interests or competencies. For the community to evolve beyond the enlistment of new contributors or volunteers, it becomes reasonable to identify project roles, people who fill such roles, and the informal procedures or rites of passage needed for participants to move between roles [cf. Fielding 1999, Kim 2000].

To mitigate against possible conflicts over responsibility, software ownership, or the quality of contributions made, the community will need to decide what kind of open source software license (cf. <http://www.opensource.org/licenses/>) should be adopted as the guideline for community property and reuse rights. A wildly successful project community becomes engaged with growing numbers of contributors. This reflects an

emerging free, OSS technology niche and the need to defend the niche from proprietary encroachments, as well as to accept tax-free donations. At this point, the community may seek to incorporate itself as a non-profit foundation to create a board of directors or oversight committee, and to assure the legal (copyright) status of the software products and community ownership.

Contributing external source code to help community bootstrapping. Many free OSSD projects like GNUenterprise.org develop an implementation of existing software system/application or functionality that exists in the marketplace. This is a form of software reinvention and integration. It enables participants to establish or expand their competency in a particular domain of software expertise. It also empowers software developers who may be participating in multiple free, OSSD projects, to integrate free software from a related domain, by adding a DBMS, project management system, or content management system to the project. This can enhance the quality and capability of the project software, as well as recognizing the contributors' gift to the community. These multi-project contributors may then become core developers, who can then assume a role in committing suggested modifications or contributions to the source code they "own", (re)configure and manage.

(Re)organizing the project team. As developers are generally participating in a free OSSD project on their own terms, there can come a time when they want to revise or abandon their community responsibilities. This gives rise to a need to solicit and designate some form of community role hierarchy and responsibility in order to identify and enable project career migration among core developers, secondary contributors, and occasional volunteers. This helps to identify opportunities for other contributors to move up or down in the community, as well as identify emerging secondary and tertiary participant roles. Frequent posting and updates to the project's to-do lists enables participants more opportunity to self-select their way through the virtual project management structure. Beyond this, other conditions that give rise for recognizing the need to reorganize the project team or role responsibilities comes through the adoption, integration, or reconfiguration of (new) community software tools, repositories, or other shared information resources.

Scanning the external software development world for innovation opportunities. One way how free, OSSD projects evolve the functionality, quality, and capability of their software products is through reinvention or innovation. Contributors bring a wealth of experience and competencies to a project, as well as their interest in many kinds of software application domains. They may import interesting concepts, techniques, or tools from other free, OSSD projects they know or have discovered. This can arise from browsing informative Web sites and other media for content on new tools, techniques, concepts, or data source that may add value to local OSSD project. They can acquire additional knowledge of these opportunities, as needed, from external publications, conferences, or informal gatherings with end-users or other software development colleagues. Based on these experiences and their expertise, the contributors may propose new features or enhancements via the local OSSD project mail list server or IM channel for how and why to incorporate the innovation opportunity. Thus, reinvention,

community debate, coding and integration are part of what drives the evolution of free, OSSD communities.

Developing and deploying community support software. Free OSSD projects both develop software of one kind, and use software of other kinds. However, there is no universal standard for what kinds of support software should be employed with free OSSD projects. However, emerging best practices for OSSD reveal that tools for code development, community communications (IM), and community code configuration management (e.g. CVS) are needed, as is a content management system to sustain and publish the community's shared information resources. Thus, it becomes necessary for participants to come forward and take responsibility for setting up and updating the project Web site/portal, and its content structure.

This is an important kind of contribution, though is not specific to the project's source code functionality, such as that for ERP or EB software modules. Similarly, the need arises for one or more participants to setup and update project resource repositories including network file servers that host project Web server and content management server, CVS directory archives, and any redistribution or "mirror" Web sites that store duplicates of current executable software release. Finally, some participant(s) must setup and post links to project specific software development tools (e.g., CVS, GNUe Forms IDE). Thus, these resources are part of the community's information infrastructure, rather than (in general) its products. So developing, deploying, and sustaining a community's project information infrastructure is critical to the ongoing evolution of the community and its software products.

Developing and maintaining the community Web site/portal. In order for a OSSD project to be recognized as such, it needs some sort of presence on the Web. Fortunately, this is increasingly a relatively simple undertaking for developers, once a networked computing infrastructure and Internet access services are provided. Such an investment and sustained operating costs may be borne by a corporate sponsor, otherwise individual community members must come forward to subsidize the project's Web presence. Once operational, the Internet domain name and related name variants for the project site are identified and registered (e.g., www.gnuenterprise.org and www.gnue.org). Similarly, project leaders will create a home page, site index, and Web site architecture for the project's content prior to announcing project to the larger OSS community. Such an announcement can be made and repeated indefinitely by registering the project's identity and Web site at SourceForge, Freshmeat, Savannah.org or similar OSSD portals. In addition, project community members can further advance recognition and awareness of the project, as well as its Web site, by posting Email messages in news or discussion groups associated with other free OSSD projects.

The GNUenterprise.org community as a capability enabling Open EC/B

As indicated earlier, many of the participants in the GNUenterprise.org project seek individual or small business opportunities to help motivate their participation in the project (cf. Figure 7). However, the project and its community may also provide a preview into how OSSD, EC, and EB may come together to give rise to Open EC/B.

Free, OSSD projects like GNUenterprise.org assign the development, sharing, learning, reuse, modification, and redistribution of free, open source code. In GNUenterprise.org, this is the GNUe software. However, what about the ERP and EB *processes* that are suppose to be supported in the user organizations that adopt and deploy the GNUe software *products*? Are these processes suppose to be closed, opaque, and patented like those for Amazon.com shown in Figure 1, or can they be open for community development, sharing, learning, reuse, modification, and redistribution? Can there be free open source processes, whether in the form of informal narratives, interactive Web-based hypermedia displays/presentations, or computational process models [Noll and Scacchi 1999, Scacchi 2001, 2002, 2002d]?

It seems that the all of the resources and capabilities that support the development of the GNUe software are much the same as might be needed to develop free, open source process models. Developing open source processes could well be a less demanding and more rewarding endeavor that would complement the free GNUe software products. The culture and beliefs manifest in the GNUenterprise.org project community suggest that the difficulties overcome in making the GNUe software development self-sustaining and eventually successful, may pale in comparison to those required to support the ERP and EB processes that the GNUe software is designed to support. Thus, it appears that the GNUenterprise.org community collectively acts in a manner that is readily aligned with a movement toward open source processes that use free, open source ERP, EB and EC software products.

Analyzing the GNUenterprise.org Case

This section presents an interpretive analysis of the case study, as is appropriate for the kinds of data and descriptions that have been presented and in related studies [cf. Scacchi 2001, 2002b, Skok and Legge 2002]

A reasonable question to ask at this point concerns whether GNUenterprise.org is an efficient and effective enterprise, and whether its participants realize gains that outweigh their individual investments. As a free OSSD alliance and virtual enterprise, GNUenterprise.org is not designed to make money or be profitable in the conventional business sense. It is, however, conceived to be able to make complex ERP and EB software modules. Companies that provide paid software developers to work on the GNUe software expect to make money from consulting, custom systems integration and deployment, and ongoing system support. These services generally accompany the installation and deployment of this kind of software. They may also just seek to acquire, use, and deploy open ERP or EB applications for their own internal EB operations. Similarly, they may value the opportunity to collaborate with other firms or other highly competent ERP and EB software developers [Crowston and Scozzi 2002, Monge, *et al.*, 1998]. Other unpaid contributors and volunteers make also share in these same kinds of values or potential outcomes.

Can an enterprise make money from creating a complex ERP and EB software suite that from the start is distributed as free, open source software? Don't ERP and EB software products whose proprietary closed source alternatives from SAP and others cost upwards of a million dollars or more [Curran and Ladd 2000, Keller and Tuefel 1998]? Yes, closed source ERP and EB systems do entail a substantial acquisition, implementation, deployment, and support costs. But the purchase price of most ERP software packages and EB service application may only represent 5-10% of the total cost of a sustain deployment in a customer enterprise. Subsequently, most of the financial cost of an ERP or EB application deployment is in providing the installation, customization, and maintenance support services. As free OSS in widespread use is subject to continuous improvement, then the opportunity to provide ongoing support services to businesses or government agencies that rely on them will continue and grow. Thus, a free OSS project like GNUenterprise.org can serve to generate opportunities for support service providers, without the need to generate revenues from sales of their ERP and EB software.

What kinds of challenges can make the transition from EC/EB to Open EC/B problematic or motivating, and how might these problems be mitigated via OSSD? Two broad categories of challenges to Open EC/B are apparent: those involving economic conditions and those denoting structural or resource-based capabilities.

Economic conditions supporting free open source software development

For Open EC/B to succeed, it must be able to overcome classic problems of "free riders" that can undermine the potential effectiveness of collective action that creates and sustains public goods [Marwell and Oliver 1993, Olson 1971, Samuelson 1954]. From an informal survey of empirical studies of OSSD to date, the public goods that OSSD participants realize is not simply source code and shared artifacts, but also *socio-technical connectivity* and *communality* [cf. Monge, *et al.* 1998]. These network externalities appear to emerge as a key variable that sustains and evolves OSSD projects like the GNUe software and the GNUenterprise.org community.

Beyond this, the GNUe software for ERP and EC/EB also offers economic or cost management alternatives including:

- *Cost-reduction* -- no direct cost to acquire OSS like GNUe modules that support ERP and other EC/B processes. Thus the direct cost of acquiring OS software is lower compared to that of commercial closed source software products.
- *Cost-avoidance* -- freedom to choose who the provider of deployment, integration, and support service providers will be, enables choice based on cost and quality of service delivered, versus the comparatively high cost structure of consulting and maintenance services provided to large ERP system customers.
- *Financial incentives (or rewards) to participate in OSSD projects* -- there is a small but growing base of evidence that suggests software developers who become core contributors to OSSD projects often earn higher salary, fees, or wages [cf. German 2002, Hann *et al.*, 2002]. Also, individuals who make their living through provision of custom consulting services also benefit from the expertise and experience with OSSD projects like GNUenterprise.org (see Figures 7 and 13).

- *Revenue enhancement and margin growth* -- Start-up firms and other small to midsize firms are emerging to address the growing market for deployment, customization, and support services based around free or low-cost OSS products, like GNUe software.
- *Capital investment and sponsorship* -- large firms like IBM, HP, DKW, and AOL continue to invest in the development and deployment of OSS products and services, and some of this investment is targeted to support small to midsize companies like RedHat, VA Software and Collab.Net.
- *Venture strategy* -- Venture capital firms were responsible for helping to create and elevate the start-up companies like RedHad, VA Linux, and Collab.Net in the late 1990's into small to midsize ventures through their capital investment. Intel, Oracle, and other firms are also investing in start-up companies whose products or service offerings are based on OSS.
- *Sustained development and growth of community capital* -- Free and open source software project communities like GNUenterprise.org create and consume many kinds of intangible assets [Millen, Fountaine, Muller 2002]. Detailing and assigning a financial value to these assets is not an easy undertaking. However, that is not to say that such assets have no significant business value. Instead, the social capital that is created and grown through an ongoing collaborative software development community like that for GNUenterprise.org is likely a key resource that binds participants to the project [cf. Preece 2000].

Thus, it appears that the economic incentives are in place that can overcome the traditional challenges to the problems of free riders created by OSSD projects like GNUenterprise.org. With this in mind, we turn to examine the resources and resource-based capabilities that enable Open EC and Open EB in a virtual organization and community project like GNUenterprise.org.

Resources and Capabilities for Open EC/B

What kinds of resources or business capabilities are needed to help make Open EC/B efforts more likely to succeed? Based on what we observed in the GNUenterprise.org case study, the following kinds of resources enable the development of both free open source ERP/EB software and community that is sustaining its evolution, application and refinement:

Personal software development tools and networking support

Free OSS developers, end-users, and other volunteers provide their own personal computing resources in order to access or participate in a free OSSD community project. They similarly provide their own access to the Internet, and may even host personal Web sites or information repositories. Furthermore, free OSS developers bring their own choice of tools and development methods to the community. The sustained commitment of personal resources helps subsidize the emergence and evolution of the community, and its shared (public) information resources. It also helps create recognizable shares of the commons that are linked (via hardware, software, and Web) to the community's information infrastructure.

Beliefs supporting Free OSSD

Why do software developers and others contribute their skill, time, and effort to the development of free OSS and related information resources? Though there are probably many diverse answers to such a question, it seems that one such answer must account for the belief in the freedom to share, learn, modify, and redistribute the evolving results from a free OSSD project. Without such belief, it seems unlikely that there could be "free" and "open source" software development projects [DiBona, Ockman and Stone, 1999, Pavlicek 2000, Williams 2002]. However, one important consideration that follows is what are the consequences from such belief, and how are these consequences put into action.

In looking across the many exhibits of data that were presented (Figures 5-21), in addition to examination of the online GNUenterprise.org information resources from which they were taken, many kinds of actions or choices emerge from the development of free OSS. Primary among them is freedom of expression and choice. Neither of these freedoms is explicitly declared, assured, or protected by free software copyright or community intellectual property rights. These additional freedoms are expressed in choices for what to develop or work on (choice of work subject or personal interest over work assignment), how to develop it (choice of method over corporate standard), and what tools to employ (personal tool choice versus only using what is provided). They also are expressed in choices for when to release work products (choice of satisfaction of work quality over schedule), determining what to review and when (modulated by community ownership responsibility), and expressing what can be said to whom with or without reservation (modulated by trust and accountability mechanisms). Shared belief and practice in these freedoms of expression and choice are part of the virtual organizational culture that characterizes a community project like GNUenterprise.org [Elliott and Scacchi 2002]. Subsequently, putting these beliefs and cultural resources into action continues to build both community and free software.

Competently skilled and self-organizing software developers

Developing complex software modules for ERP applications requires skill and expertise in the domain of EB and EC. Developing these modules in a way that enables an open architecture requires a base of prior experience in constructing open systems. The skilled use of project management tools for tracking and resolving open issues and bug reports also contributes to the development of such a system architecture. These are among the valuable professional skills that are mobilized, brought to, or drawn to free OSSD community projects like GNUenterprise.org [cf. Crowston and Scozzi 2002]. These skills are resources that free OSS developers bring to their projects.

Free OSS developers organize their work as a virtual organizational form that seems to differ from what is common to in-house, centrally managed software development projects. Within in-house development projects, software application developers and end-users often are juxtaposed in opposition to one another. Danziger [1979] referred to this concentration of software development skills, and the collective ability of an in-house development organization to control or mitigate the terms and conditions of system

development as a "skill bureaucracy". Such a software development skill bureaucracy would seem to be mostly concerned with rule-following and rationalized decision-making, perhaps as guided by a "software development methodology" and its corresponding computer-aided software engineering tool suite.

In the decentralized virtual organization of a free OSSD community like GNUenterprise.org, a "skill meritocracy" [cf. Fielding 1999] appears as an alternative to the skill bureaucracy. In such a meritocracy, there is no proprietary software development methodology or tool suite in use. Similarly, there are few explicit rules about what development tasks should be performed, who should perform, when, why, or how. Instead, free OSSD participants organize around the expertise, reputation, and accomplishments of core developers, secondary contributors, and tertiary reviewers and other volunteers.

Participants nearer the core have greater control and discretionary decision-making authority, compared to those further from the core. However, realizing such authority comes at the price of higher commitment of personal resources described above. Being able to make a decision stick or to convince other community participants as to the viability of a decision, advocacy position, issue or bug report, also requires time, effort, communication, and creation of project content to substantiate such an action. This authority also reflects developer experience as an interested end-user of the software modules being developed. Thus, developers possessing and exercising such skill may be intrinsically motivated to sustain the evolutionary development of their free open source ERP and EB software modules, so long as they are active participants in their community project.

Discretionary time and effort of developers

Are OSS developers working for "free" or for advancing their career and professional development? Following the survey results of Hars and Ou [2002], there are many personal and professional career oriented reasons for why participants will contribute their time and effort to the sometimes difficult and demanding tasks of software development. What we have found in GNUenterprise.org appears consistent with their observations. These include not only self-determination, peer recognition, community identification, and self-promotion, but also belief in inherent value of free software [cf. DiBona, Ockman, and Stone, 1999, Pavlicek 2000, Williams 2002].

In the practice of self-determination, no one has the administrative authority to tell a project member what to do, when, how, or why. OSS developers can choose to work on what interests them personally. Free OSS developers, in general, work on what they want, when they want. However, they remain somewhat accountable to the inquiries, reviews, and messages of others in the community, particularly with regard to software modules or functions for which they have declared their responsibility to maintain or manage as a core developer.

In the practice of peer recognition, a developer becomes recognized as an increasingly valued community contributor as a growing number of their contributions make their way into the core software modules [Bergquist and Ljungberg 2001]. In addition, nearly two-thirds of OSS developers work on 1-10 *additional* OSSD projects [Hars and Ou 2002], which also reflects a growing social network of alliances across multiple free, OSSD projects [cf. Monge, *et al.* 1998]. The project contributors who span multiple project communities can serve as "social gateways" that increase the community's mass [Marwell and Oliver 1993] and opportunity for inter-project software composition and bricolage. It also enables and empowers their recognition across multiple communities of free OSSD peers.

In building community identification, project participants build shared domain expertise, and identify who is expert in knowing how to do what [cf. Ackerman and Halverson 2000]. Interlinked contents and persistent communicated messages help point to who the experts and core contributors are.

In self-promotion, project participants communicate and share their experiences, perhaps from other application domains or work situations, about how to accomplish some task, or how to develop and advance through one's career [see Figures 7 and 13]. Being able to move towards the center or core of the development effort requires not only the time and effort of a contributor, but also the ability to communicate and convince others as to the value or significance of the contributions. This is necessary when a participant's contribution is being questioned in open project communications, not incorporated (or "committed") within a new build version, or rejected by vote of those already recognized as core developers [cf. Fielding 1999].

The last source of discretionary time and effort observed in GNUenterprise.org is found in the freedoms and beliefs in free OSSD that are shared, reiterated and put into observable interactions. If a community participant fails to sustain or reiterate the freedoms and beliefs institutionalized in the GPL, then it is likely the person will leave the project and community. But understanding how these freedoms and beliefs are put into action points to another class of (sentimental) resources that must be mobilized and brought to bear in order to both develop free OSS systems and the global communities that surround and empower them.

Trust and social accountability mechanisms

Developing complex software modules for ERP, EB, or EC applications requires trust and accountability among project participants. Though trust and accountability in a free OSSD project may be invisible resources, ongoing software and community development work occur only when these intangible resources and mechanisms for social control are present [cf. Gallivan 2001, Hertzum 2002].

The intangible resources arise in many forms. They include assuming ownership or responsibility of a community software module, voting on the approval of individual action or contribution to community software [Fielding 1999], shared peer reviewing

[DiBona, Ockman and Stone 1999], and by contributing gifts [Bergquist and Ljungberg 2001] that are reusable and modifiable public goods [Olsen 1971, Samuelson 1954]. They also exist through the community's recognition of a core developer's status, reputation, and geek fame [Pavlicek 2000]. Without these attributions, developers may lack the credibility they need to bring conflicts over how best to proceed to some accommodating resolution. Finally, as a free OSSD project grows in terms of the number of contributing developers, end-users, and external sponsors, then community's mass becomes sufficient to insure that individual trust and accountability to the project community are sustained and evolving [Marwell and Oliver 1993].

Thus, free OSSD efforts rely on mechanisms and conditions for gentle but sufficient social control that helps constrain the overall complexity of the project. These constraints act in lieu of an explicit administrative authority or project management regime that would schedule, budget, staff, and control the project's development trajectory with varying degrees of administrative authority and technical competence.

Free open source software development informalisms

Software informalisms [Scacchi 2002a] are the information resources and artifacts that participants use to describe, proscribe, or prescribe what's happening in a free OSSD project. They are informal resources that are comparatively easy to use, and immediately familiar to those who want to join the community project. However, the contents they embody require extensive review and comprehension by a developer before core contributions can be made. The most common informalisms include community communications and messages within Email, threaded Email discussion forum, news postings, community digests, and instant messaging chat. They also include scenarios of usage as linked Web pages, how-to guides, to-do lists, FAQs, and other itemized lists, as well as traditional system documentation and external publications. Free OSS community property licenses also help to define what software or related project content are protected resources that can subsequently be shared, examined, modified, and redistributed. Finally, open software architectural designs, scripting languages like Perl and PHP, and the ability to either plug-in or integrate software modules from other OSSD efforts, are all resources that are used informally, where or when needed according to the interests or actions of project participants.

All of the software informalisms are found or accessed from project related Web sites or portals. These Web environments are also software informalisms [Scacchi 2002a].

A Web presence helps make visible the community's information infrastructure and the array of information resources that populate it. These include OSSD community project Web sites (e.g., SourceForge.net, Savannah.org, and Freshment.org), community software Web sites (PHP-Nuke.org), and project Web site (www.GNUenterprise.org), as well as embedded project source code Webs (directories), project repositories (CVS), and software bug reports and issue tracking data base (DCL).

Together, these software informalisms constitute a substantial collection of information resources and artifacts that are produced, used, consumed, or reused within and across free OSSD projects.

Free OSSD capability enabling free, open ERP and EB systems

The array of social, technological, and informational resources that enable a free OSSD project is substantial. However, they differ in kind and form from the traditional enterprise resources that are provided to support proprietary, closed source software systems. These traditional resources are money (budget), time (schedule), skilled development staff, project managers (administrative authority), quality assurance (QA) and testing groups, documentation writers, computer hardware and network maintainers, and others. Free OSSD projects seem to get by with comparatively small amounts of money, though subsidies of various kinds and sources are present and necessary. They also get by without explicit schedules, though larger projects may announce target release dates, as well as (partially) order which system functions or features will be included in some upcoming versions, for some target release. Further, they get by without the rule-making and decision-making authority of project managers, who may or may not be adept at empowering, coaching, or rewarding development staff to achieve corporate software development goals. The remaining resources are provided within a free OSSD effort via subsidies, sponsorship, or volunteer effort.

Thus, the resources for free OSSD efforts are different: they are not mobilized, allocated, or otherwise brought to bear in the manner traditional to the development of proprietary, closed source software systems. Hopefully, it should be clear that the differences being highlighted are not based simply on a comparison of functionality or features visible in the development or use of open vs. close source software products. As such, the resource-based capability for developing free OSS components or modules for ERP, EB and EC applications is different.

Revisiting the Research Questions

With the preceding analysis in mind, we now turn to identify the research questions that are examined through the case study of GNUenterprise.org. These questions serve to explicate how open source software development and EC/EB approaches might be brought together, and to what ends.

How are free OSSD processes similar to and different from EC or EB application development and deployment processes? They are similar in that they generally span multiple organizations. However, they differ in that OSSD processes occur within a virtual organization [Crowston and Scozzi 2002, Noll and Scacchi 1999, 2001] that lacks a single administrative authority to coordinate, schedule, and provides the resources necessary to sustain the development effort. They are similar in that most existing as-is or legacy business processes or product development workflows are poorly understood by the people who routinely perform those processes. They differ in that OSSD processes have the potential to be codified using free open source process modeling notations [Noll

and Scacchi 2001, Scacchi 2002d], whereas existing EC or EB processes tend to be closed, proprietary, and potentially patented (see Figure 1). They differ in that OSSD efforts employ decentralized management of processes and system architecture, while their EC and EB counterparts are generally centrally managed. They also differ in that OSSD efforts rely on self-elected work assignments, whereas EC and EB development and deployment rely on explicit staffing assignments or delegations. Last, they differ in that OSSD efforts are centered around ongoing development of global community and free software, while EC or EB development efforts are centered about development of corporate goals and proprietary closed source products or services.

What is involved in designing EC or EB processes to take advantage of OSSD processes, practices, and infrastructure? First, it is necessary to recognize many ERP or EB product similarities may be apparent, while process similarities are few. It is apparent that the processes that surround the development, deployment, use, and ongoing support of EC, EB, or OSSD development are underspecified or unexamined. Without knowledge of these processes, it seems likely that the deployment, use, or support of EC or EB software will be problematic. In contrast, acquiring knowledge of these processes, and putting them into practice, is recognized as an appropriate technique that increases the likelihood of a successful and sustainable ERP system implementation [Keller and Tuefel 1998]. Second, taking advantage of OSSD processes, practices, and information infrastructure seems to depend on adopting and adapting decentralized self-elected workflow and process management, together with the information resources that support them.

What kinds of challenges make the transition from EC/EB to Open EC/B problematic, and how might these problems be mitigated? First, Open EC/B depends on a capability to foster both the development of a (relatively small) global community and free open source software components. The risks associated with global community development, at least as manifest at SAP, SUN (see Figure 3), or HP (see Figure 4), seem to be fairly modest, while not incurring significant costs [cf. Dinkelacker, Garg, *et al.*, 2002]. Their experiences may therefore be suggested for further consideration or replication. However, the experience of companies like SAP suggests that taking an existing closed source software product that has had little to moderate commercial success, and making it free open source software, does not guarantee that a global community will emerge to further its evolution or deployment.

Second, common EC or EB system development projects rely on traditional, centrally managed software project management tools and techniques. OSSD projects rely on fostering and propagating beliefs in software development freedoms, as well as collaborative, community oriented software development. OSSD projects rely on a meritocracy of competently skilled and professionally motivated software developers who use their own tools, rather than a bureaucracy of software developers who must be managed to conform to standard development processes, plans, and application interfaces using corporate provided tools.

Third, Open EC/B projects seem to depend on emergence and support of a global community of developers, end-users, and affiliated sponsors. Community development

can be cultivated and thrive within corporate boundaries, perhaps through building up corporate knowledge management initiatives that focus on recognizing and enabling communities of practice for OSSD, internal OSSD style pilot projects and OSSD infrastructure deployment. HP [Dinkelacker, Garg, *et al.*, 2002], IBM, SUN, DKW, and BGI have all invested in these kinds of efforts.

What kinds of resources or business capabilities are needed to help make Open EC/B efforts more likely to succeed? The resources that are mobilized to enable and sustain free OSSD and Open EC/B efforts include first, the participants' personal software development and networking infrastructure. Second, it includes the career interests, base of skills and expertise, time, effort, beliefs and freedom of expression and choice that developers want, in order to further their individual or community advancement. Third, it includes the software informalisms that represent and embody the content, workspace, and workplace of Open EC/B development efforts. These resources must be mobilized by free OSSD project participants, in order to enable both the resources and capabilities for software and community development.

Implications and Enterprise Process (Re)Design Heuristics

Four categories of implications and enterprise redesign suggestions are identified in this section.

First, OSSD is being applied by commercial firms of all sizes to create, deploy, and sustain free ERP systems, and other EC/EB systems. OpenEC/B efforts are emerging and being positioned to compete with high-cost ERP systems or EB applications. The resulting application systems or service offerings can be marketed to small-to-medium size firms. These firms are generally outside the market addressed by large ERP software providers like SAP, Baan, Oracle, and others. In contrast, companies and independent consultants participating in GNUenterprise.org effort expect to profit from the offering of installation, consulting, training, and ongoing support services to firms that want high performance, low cost, reliable, and adaptable ERP capabilities for EC/EB operations. Free, open source ERP systems are envisioned to be enable such opportunities, as was explained through the case study and analysis. This suggests an opportunity may exist for companies that are planning to embark on a new EC or EB corporate initiative to consider how Open EC/B processes may be employed and supported by free open source ERP or EB software modules. Though the emerging market for GNUenterprise.org software may be targeted at small or midsize firms, the GNUe software should provide a significant starting point for evaluating and redesigning a company's legacy or new EB processes around the use of ERP or EB functionality provided in the GNUe software.

Second, OpenEC/B can arise within an international virtual organization. There is no single company in charge of, directing, or providing the resources for this effort. This is not simply a sub-contracted or outsourced system development effort. It is instead an effort to create a community of individual contributors and commercial firms who can profit (financially or professionally) from the development, deployment, and support of free, open source ERP systems and related EC/EB capabilities. A growing number of

large established firms, as well as many small, start-up firms have begun to explore how to embrace or support OpenEC/B via virtual organizations, as identified in the case study report. One enterprise process opportunity that this suggests is for a company to contribute developers who can download and evaluate the GNUe software. Focus here could be directed at providing feedback and assessment to the GNUenterprise.org community regarding what kinds of functionality or ERP or EB process support would make the GNUe software a more effective for deployment within a large company. Large companies like Boeing have experience in virtual enterprises, but not those employing or relying on Open EC/B software or processes. Thus, the opportunity exist for large companies to participate in a virtual organization using Open EC/B products and processes.

Third, OpenEC/B services are being deployed and marketed as new business activity. Further, we are finding tractable resource dependencies exist that shape OpenEC/B organizational systems and process (re)design. The implications of OpenEC/B services as a business model are explored in the case study and analysis. However, the current focus of Open EC/B services are those targeted to small and midsize firms that are likely to engage small firms or independent consultants as the service providers. Clearly, large firms can potentially identify an internal business unit that might resemble or work like a small or midsize firm. Such an internal business unit can therefore serve as an in-house pilot study that can experimentally engage and evaluate how Open EC/B software and processes might be designed and deployed as a technique for realizing the economic benefits of free open source software for ERP and EB applications.

Fourth, OSSD is being applied by commercial firms to create, deploy, and sustain free, open source Enterprise Resource Planning (ERP) systems, and other E-Commerce/E-Business systems. Established large firms developing these kinds of software systems like SAP and Oracle are venturing into the development, use, or provision of open source software products or platforms. While such ventures may simply be testing the waters of the marketplace, they do have the potential to convey to prospective customers what the economic benefits of free OSSD efforts can be. Similarly, these ventures may represent comparatively low-cost strategic investments to see how competing firms may react or not to products that employ open source software as a competitive product or service offering.

Overall, different considerations will motivate a company of any size to explore or invest in free open source software that enables informal Open EC/B experiments or systematic evaluations. The purpose of such investigations may therefore be to determine:

- how a company with little or no prior experience with free OSSD projects might get involved in sponsoring, participating, or putting into practice such a collaborative software development effort
- how to evaluate the business value of Open EC/B software and processes
- how to setup a free OSSD project
- how to create new, or expand existing, product lines for non-IT products to incorporate free open source software capabilities

- how to create new, or expand existing, service offerings for complex products through adoption and tailoring of an free open source software development process and support community
- what economic benefits of Open EC/B can be most readily attained from deploying free Open EC/B software like that from GNUenterprise.org, and how to evaluate the costs of achieving such benefits

Conclusions

Four main conclusions can be drawn from the study, data, and analysis presented in this report.

First, we have identified and introduced a new concept called *OpenEC/B*. OpenEC/B denotes the integration of free open source software development resources, products, and processes, with the existing or emerging capabilities for Electronic Commerce/Business. This concept is introduced and its consequences are explained in the case study and analysis. Companies like IBM, SUN and Intel have begun to offer products that incorporate open source software systems. Companies like Microsoft have chosen not to offer such products, but to instead offer non-free "shared" access to proprietary source code subject to non-disclosure agreements, and to discourage open source software practices. Thus companies like IBM and SUN may see that open source software systems may serve as part of a competitive strategy against companies like Microsoft and others that seek to exert non-free access and control over their proprietary software products. For other companies like Canon, Quantum/ATL, or Boeing, they may face a strategic choice for whether or not to embrace open source systems or Open EC/B processes for internal use, or for external product offerings, in their respective markets. Similarly, companies like Conexant or Intel will face similar choices as open source capabilities move further into the realm of high-volume consumer products with embedded system components.

Second, this study is the *first* to link OSSD with EC/EB. No prior case studies of EC/EB have identified or addressed whether or how open source software methods might be applied or integrated with EC/EB. Similarly, there is no prior study of how commercial firms expect to profit from investments in open source software for EC/EB applications, or from reduced cost of ownership of critical EC/EB systems built from free, open source system components. Thus, there is an opportunity for CRITO Consortium member firms to begin considering whether these results merit timely consideration or exploratory investments. For example, companies offering consumer products (e.g., Canon, Intel) or high value, information technology based products and services (e.g., Boeing, IBM, Quantum/ATL, or DoD) may begin to consider whether open source of OpenEC/B capabilities that offer lower purchase prices, lower total cost of ownership, and higher quality [Scacchi 2002c] represent new market entry or new product differentiation opportunities.

Third, we have identified resources and resource-based capability for OpenEC/B that may explain/predict (a) what's involved, (b) how it works, or (c) what conditions may shape the longer-term success or failure of such efforts. In simple terms, these resources

include time, skill, effort, belief, personal and corporate subsidies, and community building on the part of those contributing as developers and users of OpenEC/B systems and techniques. Of these, *belief* in the freedoms that open source system development allows [Elliott and Scacchi 2002] appears central. Developers and users who believe in the promise and potential of OpenEC/B systems are willing to allocate (or volunteer) their time and apply their skills to make the effort of developing or using open source systems a viable and successful course of action. Thus companies seeking to invest in or exploit OpenEC/B techniques or systems must account for how it can most effectively cultivate an OpenEC/B culture, belief system, and community of practice, as part of their strategic choice. IBM's *developerWorks*, SUN's *Java Community Process*, and HP's progressive source initiative represent initial efforts in this direction.

Last, we have identified a set of large, midsize, and small companies pursuing OpenEC/B business models. The large companies in the IT hardware and software markets are experimenting with market initiatives whose purpose seems to be to determine what the market is or can be for open source software based products or solutions. They also seek to determine whether these market initiatives represent a viable competitive strategy for opposing Microsoft, which is a staunch supporter of closed, proprietary software systems. Large companies not involved in the IT industry are making strategic investments in open source software development in an attempt to find newer, faster, cheaper, and better solutions to problems they face when integrating complex applications from multiple vendors for use in a global E-Business environment. Midsize and small companies involved in OSSD primarily seek to expand their business service offerings and revenue through support and consulting service contracts. The opportunity for these companies appears mainly in their ability to address the needs of customers who also represent small or mid-sized businesses that seek low-cost, incremental investment kinds of approaches to IT-based systems for EC or EB applications. The ability of these emerging OSSD-based service providers thus remains to be demonstrated and shown viable over the long term.

References

M. Ackerman and C. Halverson, Reexamining Organizational Memory, *Communications ACM*, 43(1), 59-64, January 2000.

M. Bergquist and J. Ljungberg, The Power of Gifts: Organizing Social Relationships in Open Source Communities, *Information Systems J.*, 11(4), 305-320, 2001.

E. Castronova, *Virtual Worlds: A First-Hand Account of Market and Society on the Cyberian Frontier*, The Gruter Institute Working Papers on Law, Economics, and Evolutionary Biology, Vol. 2, Article 1, December 2001.
<http://www.bepress.com/giwp/default/vol2/iss1/art1>

F. Cortiana, "Norms in matter of computer science pluralismo on the adoption and the spread of the free software and on the portabilità of documents informed to us in Public Administration" (XIV Legislatura Action Senate n. 1188) (in Italian) April 2002.
<http://www.softwarelibero.it/altri/cortiana.shtml>

K. Crowston and B. Scozzi, Open Source Software Projects as Virtual Organizations: Competency Rallying for Software Development, *IEE Proceedings--Software*, 149(2), 3-17, 2002.

T.A. Curran and A. Ladd, *SAP R/3 Business Blueprint: Understanding Enterprise Supply Chain Management* (2nd. Edition), Prentice-Hall, Upper Saddle River, NJ, 2000.

M. Cusumano and D.B. Yoffie, Software Development on Internet Time, *Computer*, 32(10), 60-69, October 1999.

CW360, *ComputerWeekly*, JD Edwards pushes modular ERP, 12 June 2002.
<http://www.cw360.com/bin/bladerunner?REQUNIQ=1025733553&REQSESS=SfGFI2G7&REQHOST=sitel&REQAUTH=2101053&2131REQEVENT=&CFLAV=1&CCAT=2&CCHAN=24&CARTI=113205>

J. Danziger, The Skill Bureaucracy and Intraorganizational Control: The Case of the Data-Processing Unit, *Sociology of Work and Occupations*, 21(3), 206-218, 1979.

B. Deltor, The Corporate Portal as Information Infrastructure: Towards a Framework for Portal Design, *Intern. J. Information Management*, 20(2), 91-101, 2000.

C. DiBona, S. Ockman and M. Stone, *Open Sources: Voices from the Open Source Revolution*, O'Reilly Press, 1999.

J. Dinkelacker, P.K. Garg, R. Miller, and D. Nelson, Progressive Open Source, *Proc. 24th International Conference on Software Engineering (ICSE 2002)*, Orlando, FL, ACM Press, May 2002.

- M. Elliott and W. Scacchi, Communicating and Mitigating Conflict in Open Source Software Development Projects, (submitted for publication), April 2002.
- J. Feller and B. Fitzgerald, *Understanding Open Source Software Development*, Addison-Wesley, NY, 2002.
- R. Fielding, Shared Leadership in the Apache Project, *Communications ACM*, 42(4), 42-43, 1999.
- K. Fogel, *Supporting Open Source Development with CVS*, Coriolis Press, Scottsdale, AZ, 1999.
- C. Freericks, Open Source Standards on Software Process: A Practical Approach, *IEEE Communications Magazine*, 116-123, April 2001.
- M. Gallivan, Striking a balance between trust and control in a virtual organization: a content analysis of open source software case studies, *Information Systems J.*, 11(4), 277-304, 2001.
- D.M. German, The Evolution of the GNOME Project, *Proc. 2nd Workshop on Open Source Software Engineering*, Orlando, FL, May 2002.
- I-H. Hann, J. Roberts, S. Slaughter, and R. Fielding, Why Do Developers Contribute to Open Source Projects? First Evidence of Economic Incentives, *Proc. 2nd Workshop on Open Source Software Engineering*, Orlando, FL, May 2002.
- A. Hars and S. Ou, Working for Free? Motivations for Participating in Open-Source Projects, *Intern. J. Electronic Commerce*, 6(3), 25-39, 2002.
- J.D. Herbsleb and R. Grinter, Splitting the Organization and Integrating the Code: Conway's Law Revisited *Proc. 21th International Conference on Software Engineering*, ACM Press, Los Angeles, CA, 85-95, 1999.
- M. Hertzum, The importance of trust in software engineers' assessment and choice of information sources, *Information and Organization*, 12(1), 1-18, 2002.
- N. Jorgensen, 2001, Putting it All in the Trunk: Incremental Software Development in the FreeBSD Open Source Project, *Information Systems J.*, 11(4), October 2001.
- G. Keller and T. Teufel, *SAP R/3 Process Oriented Implementation: Iterative Process Prototyping*, (translated by A. Weinland), Addison Wesley Longman, Harlow, England, 1998.
- S. Koch and G. Schneider 2001, Effort, Cooperation and Coordination in an Open Source Software Project: GNOME, *Information Systems J.*, 12(1), 27-42, January 2002.

B. Kogut and A. Metiu, Open Source Software Development and Distributed Innovation, *Oxford Review of Economic Policy*, 17(2), 248-264, 2001.

X. Li and W. Scacchi, *A Formal Architecture for Open Systems*, May 2002.
<http://www.ics.uci.edu/~wscacchi/SA/ArchitecturalDesign/Concepts/OpenSystemsArchitecture-XML/>

A. MacCormack, R. Verganti, and M. Iansiti, Developing Products on Internet Time: The Anatomy of a Flexible Development Process, *Management Science*, 47(1), January 2001.

G. Marwell and P. Oliver. *The Critical Mass in Collective Action : A Micro-Social Theory*. Cambridge University Press, 1993

D. Millen, M. Fountaine, and M. Muller, Understanding the Costs and Benefits of Communities of Practice, *Communications ACM*, 45(4), 69-74, April 2002.

A. Mockus, R. Fielding, and J. Herbsleb, Two Case Studies on Open Source Software Development: Apache and Mozilla, *ACM Trans. Software Engineering and Methodology*, to appear, 2002

A. Mockus and J. Herbsleb, Why not improve coordination in distributed software development by stealing good ideas from Open Source?, *Proc. 2nd Workshop on Open Source Software Engineering*, Orlando, FL, May 2002.

P.R. Monge, J. Fulk, M.E. Kalman, A.J. Flanagan, C. Parnassa and S. Rumsey. Production of Collective Action in Alliance-Based Interorganizational Communication and Information Systems, *Organization Science*, 9(3): 411-433, 1998.

B. Nardi, S. Whittaker, and E. Bradner, Interaction and Outeraction: Instant Messaging in Action, *Proc. Conf. Computer Supported Cooperative Work (CSCW'00)*, ACM Press, Philadelphia, PA, 79-88, 2000.

J. Noll and W. Scacchi, Supporting Software Development in Virtual Enterprises, *Journal of Digital Information*, 1(4), February 1999.

J. Noll and W. Scacchi, Specifying Process-Oriented Hypertext for Organizational Computing, *J. Network and Computer Applications*, 24(1), 39-61, 2001.

E.D.V. Nunez, *Peruvian Congressman Refutes Microsoft "Fear, Uncertainty, and Doubt" (F.U.D.) Concerning Free and Open Source Software*, 8 April 2002, (original in Spanish), http://www.opensource.org/docs/peru_and_ms.html.

M. Olson, *The Logic of Collective Action*, Harvard University Press, Cambridge, MA, 1971.

R. Pavlicek, *Embracing Insanity: Open Source Software Development*, SAMS Publishing, Indianapolis, IN, 2000.

J. Preece, *Online Communities: Designing Usability, Supporting Sociability*, John Wiley and Sons, Inc., New York, 2000.

C.R. Reis and R.P.M. Fortes, An Overview of the Software Engineering Process and Tools in the Mozilla Project, *Proc. Workshop on Open Source Software Development*, Newcastle, UK, February 2002.

P. Samuelson, The Pure Theory of Public Expenditure. *Review of Economics and Statistics*, 36:387-390, 1954.

W. Scacchi, Understanding Software Process Redesign using Modeling, Analysis and Simulation, *Software Process--Improvement and Practice*, 5(2/3), 183-195, 2000.

W. Scacchi, Redesigning Contracted Service Procurement for Internet-based Electronic Commerce: A Case Study, *J. Information Technology and Management*, 2(3), 313-334, 2001.

W. Scacchi, Process Models in Software Engineering, in J. Marciniak (ed.), *Encyclopedia of Software Engineering* (Second Edition), 993-1005, Wiley, New York, 2002.

W. Scacchi, Understanding the Social, Technological, and Policy Implications of Open Source Software Development, Presented at the *NSF Workshop on Open Source Software*, January 2002a.

W. Scacchi, Understanding the Requirements for Developing Open Source Software Systems, *IEEE Proceedings--Software*, 149(2), 24-39, 2002b.

W. Scacchi, Is Open Source Software Development Faster, Better, and Cheaper than Software Engineering?, *Proceedings 2nd. Workshop on Open Source Software Engineering*, Orlando, FL, May 2002c.

W. Scacchi, Exploring Open Software System Acquisition Processes and Architectures, Final Report, Institute for Software Research, UC Irvine, Irvine, CA, June 2002d.
<http://www.ics.uci.edu/~wscacchi/Papers/DAU/Final-Report-2002.pdf>

S.R. Schach, B. Jin, D.R. Wright, G.Z. Heller, and A.J. Offutt, Maintainability of the Linux Kernel, *IEEE Proceedings--Software*, 149(2), 18-23, 2002

S.Sharman, V. Sugurmaran, and B. Rajagopalan, A Framework for Creating Hybrid-Open Source Software Communities, *Information Systems J.*, 12(1), 7-25, 2002.

W. Skok and M. Legge, Evaluating Enterprise Resource Planning (ERP) System using an Interpretive Approach, *Knowledge and Process Management*, 9(2), 72-82, 2002.

H. Tellioglu and I. Wagner, negotiating Boundaries: Configuration Management in Software Development Teams, *Computer Supported Cooperative Work: Journal of Collaborative Computing*, 6(4), 251-274, 1997.

S. Williams, *Free as in Freedom: Richard Stallman's Crusade for Free Software*, O'Reilly Books, Sebastopol, CA, 2002.