# Advancing Economic Research on the Free and Open Source Software Mode of Production

By

**J.- M. Dalle,[1] P. A. David,[2] Rishab A. Ghosh,[3] and W. E. Steinmueller[4]**

*Université Pierre et Marie Curie (Paris VI),[1] Stanford University & Oxford Internet Institute[2]*
*University of Maastricht-MERIT[3] and University of Sussex-SPRU[4]*

First draft: 27 July 2004
Second draft: 30 September 2004
This version: 3 December 2004

---

Forthcoming in

*BUILDING OUR DIGITAL FUTURE*

*Future Economic, Social & Cultural Scenarios Based On Open Standards*

Edited by Marleen Wynants and Jan Cornelis
Vrjie Universiteit Brussels (VUB) Press, Brussels 2005

---

**ABSTRACT**

Early contributions to the academic literature on free/libre and open source software (**F/LOSS**) movements have been directed primarily at identifying the motivations that account for the sustained and often intensive involvement of many people in this non-contractual and unremunerated productive activity. This issue has been particularly prominent in economists' contributions to the literature, and it reflects a view that widespread voluntary participation in the creation of economically valuable goods that is to be distributed without charge constitutes a significant behavioral anomaly. Undoubtedly, the motivations of F/LOSS developers deserve to be studied more intensively, but not because their behaviors are unique, or historically unprecedented. In this essay we argue that other aspects of the "open source" phenomenon are just as intriguing, if not more so, and possibly are also more consequential topics for economic analysis. We describe the re-focusing and re-direction of empirical and theoretical research in an integrated international project (based at Stanford University/SIEPR) that aims at better understanding a set of less widely discussed topics: the modes of organization, governance and performance of F/LOSS development -- viewed as a collective distributed mode of production.. We discuss of the significance of tackling those questions in order to assess the potentialities of the "open source way of working" as a paradigm for a broader class of knowledge and information-goods production, and conclude with proposals for the trajectory of future research along that line.

*Contact author:* Professor Paul A. David
Department of Economics
Stanford University
Stanford CA 94305-6072
Email: pad@stanford.edu

# Advancing Economic Research on the Free and Open Source Software Mode of Production*

By

**J. -M. Dalle, P. A. David, Rishab A. Ghosh, and W. E. Steinmueller**

## I

## Re-focusing Research on "Open Source" Software -- as a Paradigm of Collective and Distributed Knowledge Production

What explains the fascination that the "open source" phenomenon seems to hold for many social scientists? Early contributions to the academic literature on free/libre and open source software (**F/LOSS** hereinafter) movements have been directed primarily at identifying the motivations that account for the sustained and often intensive involvement of many people in this non-contractual and unremunerated productive activity.[1] This issue has been particularly prominent in economists' contributions to the literature, and it reflects a view that widespread voluntary participation in the creation of economically valuable goods that is to be distributed without charge constitutes a rather significant behavioral anomaly. Anomalies are intrinsically intriguing, and their identification may serve to alert us to emerging patterns of behavior, or social organization that have considerable economic and social importance. But, while the question of the motivations of F/LOSS developers is one that undoubtedly deserves closer study, the respect(s) in which their behaviors are anomalous should be precisely described by reference to some "normal," or otherwise expected behavioral patterns. The latter exercise is likely to prove valuable in bringing into clearer focus other aspects of the "open source" phenomenon that, arguably, are even more intriguing and possible far more consequential. This essay describes the re-focusing and re-direction of economic research in order better elucidate those other, less widely discussed features, which concern F/LOSS development as a collective production mode.

As a preamble to that undertaking, however, one should try to understand why the economics literature about open source software became almost instantly preoccupied with the "puzzle" of the participants' motivations. For many who presuppose economic rationality on the part of individuals engaged in time-consuming pursuits, the fact that there were growing

* The research underlying this paper was conducted by the Stanford Institute for Economic Policy Research (SIEPR) Project on the Economics of Free and Open Source Software, with the financial support of grant awards from the National Science Foundation program on Digital Technology and Society: IIS-0112962 (2001-04) and IIS-0329259 (2003-05). The authors are grateful to the colleagues and research units at the institutions with which they are, respectively, affiliated whose individual contributions to the Project are acknowledged in the various publications and working papers referred to herein.

[1] J. Lerner and J. Tirole, "The Simple Economics of Open Source", Cambridge, MA, National Bureau of Economic Research, Working Paper 7600, 2000, and Eric von Hippel, "Horizontal innovation networks – by and for users", Cambridge, MA, MIT Sloan School of Management, Working Paper No. 4366-02. June, 2002.

communities of developers who devoted appreciable time to writing and improving code without remuneration presented an aberrant form of behavior, which was difficult to to explain. At least, it was difficult if one resisted the heterodox belief that altruism not only was widespread, but also had been gaining converts in the population. Such a reading of the facts posed the challenge of how to reconcile participation of F/LOSS activities with the main (ego-regarding) tenets of modern economists' views of the driving motivations of human actions.

A second strand followed in the early economic research literature has been to search for the secret by which the F/LOSS mode of production is able to create information-goods that compete successfully in the market against proprietary software. Moreover, that they do so not simply on the basis of their significantly lower monetary cost, but, as many partisans of F/LOSS allege, on the basis of their superior quality.[2] This framing of the research agenda resembles the first theme in projecting surprise and puzzlement about the apparently greater efficiency that these non-profit, distributed production organizations have been able to achieve in relation to major software companies engaged in "closed" and centrally directed production of the same type of commodity.[3]

It is not uncommon for investigators in a new field to "hype" the mysteries that they are about to dispel, and it is characteristic of such presentations of research that it is rare indeed for their authors to describe the supposedly baffling phenomena and then announce that they remain puzzled. But we would not go so far as to discount the sense of urgency that has been attached to unraveling the mystery of what is motivating those software developers. We share the view that the F/LOSS movements carry broader economic and social significance, and therefore deserve to be the subject of continuing, systematic, empirical and theoretical study.[4] The fact that much about this particular phenomenon continues to be poorly understood, however, is not unique; the same might well be said about other aspects of the workings of modern economies, which are no less likely to prove important for human well-being.

One might therefore be excused for pointing out that if the research attention that F/LOSS software production attracts from economists is to be rationalized simply on the grounds of the novelty and mysteriousness of the foregoing phenomena, it cannot be very well founded. The existence of F/LOSS activities on their present scale hardly is so puzzling or aberrant a development as to constitute a rationale for devoting substantial resources to studying it.

---

[2] J. -M. Dalle and Nicolas Jullien, "'Libre' software: turning fads into institutions?" Research Policy 32, 2003, pp. 1-11; D. Nichols and M. Twidale, "The Usability of Open Source Software", *First Monday* (http://firstmonday.org/issues/issue8_1/nichols/index.html), 2003, last accessed 5 February 2003, and Eric S. Raymond, *The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*. Sebastopol, CA, O'Reilly and Associates, Inc., 1999.

[3] See Steven Weber, *The Success of Open Source*, Cambridge, MA, Harvard University Press, 2004, which justifies its examination of the open source phenomenon by emphasizing the posing of "a significant challenge to Microsoft in the marketplace". On the relationship between the development of Microsoft Corporation's business strategy and its software process and product characteristics in the era preceding the U.S. government anti-trust suits of the late 1990s, see Michael A. Cusumano and Richard W. Selby, *Microsoft Secrets*, New York, The Free Press, 1995.

[4] This and the following part draw upon material from J. -M. Dalle, P.A. David, and W.E. Steinmueller, "Integrated research on the Economic Organization, Performance and Viability of OS/FS Software Development," a statement prepared for the Workshop on *Advancing the Research Agenda on Free/Open Source Software*, held in Brussels on 14 October 2002 under the sponsorship of the EC IST Program and NSF/CISE & DTS.

The emergence of co-operative modes of knowledge production among members of distributed epistemic communities who do not anticipate receiving direct remuneration for their efforts is not a recent social innovation. Among the numerous historical precursors and precedents for F/LOSS are the "invisible colleges" that appeared in the 17th century and engaged practitioners of the new experimental and mathematical approaches to scientific inquiry in western Europe. The "professionalization" of scientific research, as is well known, was a comparatively late development.[5] Nor is the superior performance of such co-operative forms of organization a novel finding: philosophers of science and epistemologists, as well as work on the economics of knowledge, have noted the superiority of co-operative knowledge-sharing as a mode of generating additions to the stock of reliable analytical and empirical propositions.[6]

It is the scale and speed of F/LOSS development work and the geographical dispersion of the participants -- rather than the voluntary nature of their contributions, that properly should be deemed "historically unprecedented". But, the modularity and immateriality that are generic characteristics of software, and the enabling effects of the advances in computer-mediated telecommunications during the past several decades, would go a long way towards accounting for those aspects of the phenomenon. Is the open source movement thereby reduced to the status of a mere epiphenomenon, another among many episodes in the unfolding computer revolution? Were that to be seen as the whole of the story, we might simply assimilate F/LOSS into the larger body of "weightless" commodities, intangible information goods whose proliferation characterizes the Age of the Internet.

Yet, in addition to all that, something more seems to be involved. In our view, what warrants the attention that F/LOSS has commanded from social scientists is its connections with three deeper, and interrelated trends that have recently become evident in modern economies First among these is the movement of information goods to center-stage as drivers of economic growth. Second is the ever more widespread use of peer-to-peer modes of conducting the distribution and utilization of information, including its re-use in creating new information-goods. These two trends are bound together and reinforced by the growing recognition that the "open" (and co-operative) conduct of knowledge production offers economic efficiencies which in general surpass those of other institutional arrangements, namely those that address the resource allocation problems posed by "public goods" by protecting secretive practices, or creating and enforcing intellectual property monopolies.[7] A third trend, which is of obvious

---

[5] For accounts of the historical conditions under which "open science" developed, see P.A. David, "Common agency contracting and the emergence of open-science institutions," *American Economic Review* 88(2), 1998, pp. 15-21; P.A. David, "Understanding the emergence of 'open-science' institutions: functionalist economics in historical context," *Industrial and Corporate Change*, 13(4), 2004, pp. 571-590; Joseph Ben-David, *Scientific Growth: Essays on the Social Organization and Ethos of Science*, edited by G. Freudenthal, Berkeley, CA, University of California Press, 1991, esp. Part II, pp. 125-210.

[6] See John M. Ziman, Reliable Knowledge, Cambridge, Cambridge University Press, 1978; Philip Kitcher, *The Advancement of Science: Science without Legend, Objectivity with Illusions*, New York, Oxford U.P., 1993; P. Dasgupta and P.A. David, "Towards a new economics of science," *Research Policy*, 23, 1994, pp.487-521.

[7] See, e.g., P. A. David, "The Economic Logic of 'Open Science' and the Balance between Private Property Rights and the Public Domain in Scientific Data and Information: A Primer," in National Research Council, *The Role of the Public Domain in Scientific Data and Information*, Washington, D.C.: National Academy Press, 2003.

practical significance for social scientists and others engaged specifically in empirical studies of the F/LOSS production mode, is the growing abundance and accessibility of quantitative material concerning the internal workings of "open epistemic communities." The kinds of data that are available for extraction and analysis by automated techniques from repositories of open source code and newsgroup archives, and from the email subscriber lists generated by F/LOSS project members themselves, also offer a rapidly widening window for research on the generic features of processes of collective discovery and invention.

A further source of motivation for undertaking to exploit this opportunity, by systematically examining the micro-economics and micro-sociology of F/LOSS development communities and the "open source way of working", springs from our interest in a conjecture about the longer-term implications of the first two among the trends just described. The open source software movement may quite possibly have "paradigm-shifting" consequences extending beyond those affecting the organizational evolution of the software industry. The rise of a decentralized and fully networked mode of creating, packaging and distributing software systems, is undoubted a challenge to the dominant organizational model of closed production and a break from the era of marketing "shrink-wrapped" proprietary software packages. Possibly it is also the precursor of a broader transformation of information-goods production and distribution. Software is, after all, but one instance in the universe of information-goods, many of which share the modular and quasi-decomposable architectural features. The latter would tend to facilitate a reorganization of production and the use of advanced computer-mediated telecommunications technologies to mobilize and co-ordinate the work of large "communities of practice" formed by weakly tied and spatially distributed individuals who were able to contribute diverse skills to the collective enterprise.

In this speculative vein, one may wonder whether the principles of organization employed by open source software projects could not also be applied to integrate existing knowledge concerning the functioning of living systems (such as the human body) by creating a computer simulation structure that would have general utility in medical education, in the design of medical devices and even – if implemented at the molecular level – in the design of pharmaceutical therapies. Might such methods prove relevant in for international partnerships aimed at education and development, the exchange of cultural information such as compilations of folklore and culinary encyclopedias, or the construction of repositories of knowledge on the world's flora and fauna? If software production is simply the first manifestation of an emerging pattern of division of labor that has not been organized effectively organized (and perhaps cannot be so organized) using traditional employment and wage relationships, it seems well worth trying to better understand the opportunities arising from, and the limits to, the innovation that the "open source way of working" represents in the organization of human cultural endeavors.

Those opportunities and constraints must surely be linked to the specific problems of forming and sustaining these largely voluntary producer-associations, a consideration that leads one back to the focal point of the early literature on the motives of the people who participate F/LOSS development work, albeit with a different research agenda in mind. Motivation, recruitment and retention of developers' efforts are likely to be affected by perceptions of the utility of a project's code to a wider population of users, and hence by the performance of the development process in dimensions such as modularity, robustness, security, frequency and

persistence of bug, etc. By examining those dimensions of quality, it would be possible in principle to characterize levels of project "output" performance (i.e., for software that is sufficiently "completed" to have been released for public use). Further, even in the absence of normal market indicators, it may be feasible also to gauge end-user relative "valuation" of various "products, by observing the comparative extent and speed of adoption of software belonging to broadly similar open source offerings. Such objective and behavioral measures of the relative "utility" of F/LOSS products might provide a useful starting-point for assessments of their contributions to improving economic welfare and human well being in society at large.[8]

The trajectory of our ongoing program of research into the organizational features of the F/LOSS phenomenon has been guided by the preceding, "formative" considerations. Its main elements and their interrelationships are described in the following part of this essay, which begins by taking up issues of resource mobilization and resource allocation in the highly decentralized, non-market-directed system of F/LOSS production. The discussion proceeds next to examine questions concerning the match between the motivating interests of developer communities, on the one hand, and, on the other hand, the needs of the final users of the software systems that are being created. Then essay's third part looks toward the future directions that research in this area may usefully pursue, considering the way that agenda may be shaped by the trajectory of technological developments and the related social organization of open source communities. The fourth and concluding discussion therefore focuses on the significance of questions concerning the social dynamics of the movement, and returns to examine further the implications of interactions between the generic features of this mode of producing digital information goods and newly emerging advanced network infrastructures and the networked computer applications they will be able to support.

## II

## An Agenda for Research on the Economics of F/LOSS Production

Proceeding from the conceptual framing of the phenomenon that has been sketched above, we have taken a rather different conceptual approach from that which has hitherto dominated the recent economics literature concerning F/LOSS. A correspondingly distinctive research strategy is being pursued at Stanford University and its academic partners in France, the Netherlands and Britain by the project on *The Economic Organization, Performance and Viability of Free and Open Source Software*.[9]

---

[8] The results of the approach suggested here might be compared with those of the an alternative procedure that our project currently is exploring, which is to measure the "commercial replacement cost" of software, using industry cost models based on empirical data for closed, proprietary production of code packages of specified size, language and reliability (as measured by post beta-test bug report frequencies).

[9] This project has been supported by NSF Grants (IIS-0112962 and IIS-032959) to the Stanford Institute for Economic Policy Research's "Knowledge Networks and Institutions for Innovation Program," led by Paul David. The three associated groups in Europe are led, respectively, by Jean-Michel Dalle (University of Paris VI), Rishab Ghosh (University of Maastricht-MERIT/Infonomics Institute), and W. Edward Steinmueller (SPRU-University of Sussex). Further details and working papers from this project are available at: http://siepr.stanford.edu/programs/OpenSoftware_David/OS_Project_Funded_Announcmt.htm. This collaboration sometimes refers to itself as the Network on Software Technology Research Advances (NOSTRA). A wit,

Many of the researchers associated with our project come to this particular subject-matter from the perspective formed by previous and on-going work in "the new economics of science". Their research in that connection has been directed to questions about the organization of collaborative inquiry in the "open-science" mode, the behavioral norms and reinforcing reward systems that structure the allocation of resources, and the relationships of these self-organizing and relatively autonomous epistemic communities with their patrons and sponsors in the public and private sectors.[10] As economists looking at F/LOSS communities, the interrelated central pair of questions that remain of prime interest for us is both simple and predictably familiar. First, how do F/LOSS projects mobilize the resources, allocate the expertise and retain the commitment of their members? Secondly, how fully do the products of these essentially self-directed efforts meet the long-term needs of software users in the wider society, rather than simply providing satisfaction of various kinds for the developers?

In other words, we have begun by setting ourselves research tasks in regard to F/LOSS that address the classic economic questions of whether and how it is possible for a decentralized system of decision-making concerning resource allocation to achieve coherent and socially efficient outcomes. What makes the problem especially interesting in this case is the possibility that the institutions developed by the F/LOSS movements enable them to accomplish that outcome without help either from the "invisible hand" of the market mechanism by price signals/incentives, or from the "visible hands" of centralized managerial hierarchies. To respond to this challenge the analysis must be directed towards providing a basis for evaluating the social optimality properties of the way "open science", "open source" and kindred co-operative communities organize the production and regulate the quality of the "information tools" and "information goods" that will be used not only for their own, internal purposes, but also by others with quite different purposes in society at large.

The parallels with the phenomenon of "open science" suggests a need for a framework that is capable of integrating theories of the micro-level incentives and social norms that structure the allocation of developers' efforts within particular projects and that govern the publication of the resulting outputs as periodic "releases" of code. Theories about why researchers choose to focus on particular lines of research, and why they publish their results, provide a starting-point for examining which open source projects receive developers' attention and how these communities of developers reach decisions about the publication (i.e., release) of their work. The recognition that all systems, even very large ones are bounded also suggests a system-wide analysis. For example, general equilibrium economics tells us that we should be asking how efforts within projects are related to the mechanisms that allocate the total (even if

---

remarking on the fact that Project NOSTRA appears to be the creation of a "community of open science analysts," has asked whether the authors and their research colleagues will soon be adding "COSA" as part of their collective acronymic identity.

[10] For an introduction to research on the economics of "open science" see, e.g., P. Dasgupta and P.A. David, "Towards a new economics of science," *Research Policy*, 23, 1994, pp. 487-521; P.A. David, "The Economic Logic of 'Open Science' and the Balance between Private Property Rights and the Public Domain in Scientific Data and Information: A Primer," Ch. 4, in *The Role of Scientific and Technical Data in Information in the Public Domain: Proceedings of a Symposium,* J.M. Esanu and P.F. Uhlir (eds), Washington, D.C., The National Academies Press, 2003. [Available at: http://www.nap.edu.]

expanding) resources of the larger community among different concurrent projects, and directing the attention of individuals to successive projects, including investment in the formation of particular capabilities and sub-specialties by members of those communities. Obviously, those capabilities provide "spill-overs" to other areas of endeavor – including the production of software goods and services by commercial suppliers. It follows that to fully understand the dynamics of the F/LOSS mode, and its interactions with the rest of the information technology sector, one cannot treat the expertise of the software development community as a given, an exogenously determined resource.

### *Implementing the Organizational Economics Approach*

In implementing the approach just outlined, four lines of complementary investigation are being undertaken by our collective research effort, three of them directed to expanding the empirical base for the analysis of distinct aspects of the micro- and meso-level workings of F/LOSS communities. The fourth is integrative and more speculative, as it is organized around the development of a stochastic simulation structure designed to show the connections between the micro- and macro-level performance properties of the larger system of software production. The three areas of empirical study, along with findings from other such inquiries, are expected to provide distributions of observations which a properly specified and parameterized simulation model should be capable of simulating; whereas, reciprocally, the simulation model is intended to provide insights into the processes that may be responsible for generating patterns of the kind that are observed, and to allow an investigation into the counterfactual conditions that various policy actions would create. Thus, although these lines of inquiry can advance in parallel, their interactions are iterative: the empirical studies guide the specification of the simulation structure that is to be used to investigate their broader, systemic implications.

The initial thrust of these four complementary research "salients" can now be described briefly, taking them in turn:

- ***Distribution of developer efforts within software projects:***

The information extracted from code repositories should eventually provide robust answers to the following array of questions, which give the flavor of a major group of micro-level allocation issues that this line of inquiry is designed to address. Is the leftwards skew in the frequency distribution of contributions to the Linux kernel (i.e., the fact that relatively few contributors are responsible for a disproportionately large share of all contributions) also a feature of the distributions found to hold for the modules within the kernel? Does this hold equally for other large and complex projects? Or, putting that question another way, is the pattern of concentration in self-identified F/LOSS "authorship" one that arises from a general "power law" distribution? Alternatively, is the concentration significantly greater for some components than for others – raising questions about how efforts are directed or re-directed to achieve a higher or lower intensity of contribution? Are these distributions stationary throughout the life of the project, or does concentration grow (or diminish) over time (the former having been found to be the case for the distribution of scientific authorship in specific fields over the lives of cohorts of researchers publishing in that field)?

Micro-level resource allocation processes governing the allocation of developer efforts within software projects can be studied quantitatively by tracking the authorship distributions found in specific projects over time. A start is being made by examining an atypical yet very important and emblematic F/LOSS product: the Linux Kernel, the successive releases of which constitute a very large database containing over 3 billion lines of code. The data production work – referred to by the acronym LICKS (Linux: Chronology of the Kernel Sources) is being conducted by Rishab A. Ghosh and his colleagues at MERIT/Infonomics. It has significantly advanced the state of basic data: first, by identifying the distinctive packages of code (or "modules") within the evolving Linux kernel, and secondly, by extracting the code for successive versions and linking the dated code (contributed by each identified author, along with the components to which it relates), so that dynamic analyses of code evolution become feasible. The resulting dataset is providing a basis both for subsequent studies of the dynamics of the participation of the population of all contributors to the Linux kernel, and their patterns of co-participation across the modules, as well as the chronology of the development of the major components of the code for the operating system. In addition, this line of research is providing measures of evolving structure and the degree of technical dependence among the "modules" that form the Linux kernel.[11]

Using data on the technical features (e.g., size, technical dependency structure) of the modules forming the Linux kernel and the distributions of authorship credits, measured by the fraction of signed "commits" in each of the modules in a given release (version) of the kernel, it is possible to estimate the equations of an econometric model of code-signing and participation behaviours and draw statistical inferences from the results about the factors that influence the distribution of developers' code-writing efforts within this large, emblematic project.[12]

In addition, it has been found to be quite feasible to identify clusters of authors who work together within and across different components of the Linux kernel project; to trace whether these "clusters" grow by accretion of individuals or coalesce through mergers; and to learn whether, if they do not grow, they contract or remain constant. Further, by correlating the clusters of authors with the data on the dependence of code sections, it may be possible to obtain characterizations of the nature of "knowledge flows" between identified groups.

An important methodological issue in this line of research is to ascertain whether or not there are significant biases in the ability of the extraction algorithm to identify the distribution of

---

[11] For the methodology developed for this investigation, see R. A. Ghosh, Clustering and Dependencies in Free/Open Software Development: Methodology and Preliminary Analysis, *SIEPR-Project Nostra Working Paper*, February 2003 [available at: http://siepr.stanford.edu/programs/Opensoftware_David/Clustering and Dependencies.html]; R. A. Ghosh and P. A. David, The Nature and Composition of the Linux Developer Community: A Dynamic Analysis, SIEPR-Project Nostra Working Paper, March 2003 [available at: http://siepr.stanford.edu/programs/OpenSoftware_David/NSFOSF_Publications.html].

[12] This approach has been pursued as a first step towards more complex quantitative analyses that will exploit the self-revealing ontological features of open source software, and the array of new tools for machine extraction of data from code repositories. See J.-M. Dalle, P. A. David, R. A. Ghosh, & F. A.Wolak, "Free & Open Source Software Developers and 'the Economy of Regard': Participation and Code-Signing in the Modules of the Linux Kernel," *SIEPR-Nostra Working Paper*, June 2004**,** [available at: http://siepr.stanford.edu/programs/OpenSoftware_David/the_Economy_of_Regard.html].

authorship in this particular dataset, for which the algorithm was designed. Inasmuch as one cannot treat the Linux Kernel as a "representative" F/LOSS project, other projects, which may differ in their conventions with regard to the self-identification of contributions in the code itself, are likely to require extensions or modification of the foregoing technique of data extraction and analysis. Tools to permit the study of archival repositories of the open source codes created by concurrent version systems (e.g., CVS, Bit-Keeper), and kindred dynamic database management and annotation systems, are being developed and tested by an emerging community of "open source/libre source" software engineers with whom we have been engaged in active trans-disciplinary collaboration.[13]

- ***Allocation of developer communities' efforts among projects:***

The SourceForge site[14] contains data on a vast number of ongoing projects, including both successful and failing ones.[16] Taking the project as the unit of observation, this data provides an evidentiary basis for seeking to establish statistically the set of characteristics that are particularly influential in determining whether or not a project meets one or more of the criteria that are taken to define "success". The latter can be measured in terms of the delivery of versions of the software at various stages of completion, continued participation by players, or the citation of the project's outputs in various F/LOSS forums. Taking it as our hypothesis that software projects are most likely to achieve a number of these objectives when they are able to align a "critical mass" of adherents and develop a self-reinforcing momentum of growth, the empirical challenge is to identify the combinations of early characteristics that statistically predict the attainment of "critical mass". A supply-driven approach to the question would interpret the "community alignment" problem as one of recruiting individuals who share a belief in the efficacy of the F/LOSS mode of developing software, the diversity of their own particular interests and motives for joining the project notwithstanding; and who collectively possess the mix of differentiated skills that are needed for the work of rapidly designing, programming, debugging and upgrading early releases of the code.

Both large- and small-scale analysis seem feasible as a way of pinpointing the characteristics that enable (or fail to enable) the creation of "burgeoning" communities that propel the growth of open source projects towards critical mass and into the phase of self-catalyzing dynamics. SourceForge itself provides sufficient information about the initial features

---

[13] See Jesus Gonzalez-Barahona and Gregorio Robles, "Free Software Engineering: A Field to Explore," *Upgrade*, IV(4), August 2003; Gregorio Robles, Stefan Koch and Jesus Gonzalez-Barahona, "Remote Analysis and Measurement by Means of the CVSAnalY Tool", Working Paper, Informatics Department, Universidad Rey Juan Carlos, (June) 2004 [available at: http://opensource.mit.edu/papers/robles-koch-barahona_cvsanaly.pdf; Jesus Gonzalez-Barahona and Gregorio Robles, "Getting the Global Picture", a presentation at the Oxford Workshop on Libre Software (OWLS), Oxford Internet Institute, 25-26 June 2004. [Available at: http://www.oii.ox.ac.uk/fiveowlsgohoot/postevent/Barahona&Robles_OWLS-slides.pdf].

[14] The SourceForge.net website contains data on 33,814 open source projects, including their technical nature, intended audiences and stage of development. Records of their past history and the involvement of members of the F/LOSS community in their initiation, growth and improvement, also are available.

[16] The success/failure of a project can be defined in terms of its rate of development and the involvement (or lack of it) by the community of developers in its improvement/growth.

of projects to make it possible to analyze the influence of factors such as technical nature, intended users/audiences, internal project organization, release policies, and legal aspects (e.g., projected form of licensing).

Timing and path-dependencies may be hypothesized to affect the success or failure of F/LOSS projects, and it may be important to recognize that success or failure is not determined in isolation from the characteristics of other projects that may be competing for developers' attention. A population ecology perspective therefore may be fruitful in this connection, and for that reason interactions between the characteristics of the project and the features of the "niche" into which it is launched are being empirically investigated. Given that "developer mind-share" is limited, we may suppose that older projects are entrenched through technological lock-in processes that make it more difficult to engage the community in competing/similar ones.[17] Developers will tend to increase their co-operative activities in these older projects as they gain in experience and knowledge about them (these individuals are moving up project-specific learning curves, as well as acquiring generic and more transferable skills). Their attention to, and willingness to co-operate in other/new projects would therefore tend to decline.[18]

This kind of externality effect, through which accidents of timing and initial momentum may serve to "lock in" some projects, while locking-out others that are technically or socially more promising if considered on their intrinsic merits, has been identified in studies of the (competitive) commercial development and distribution of other technological artifacts. It would therefore be of considerable interest to establish whether or not dynamics of this kind can be observed in the non-market allocation of developmental resources among software systems products. The fact that SourceForge makes it possible to filter projects according to the tools (such as programming languages and techniques) used in their development, and that the differences between these tools may be an important factor in lock-in, makes the analysis of this kind of processes easier. The possibility of tracking down the history of individuals' co-operative activities may also make it feasible to study their involvement, entry and exit patterns in different projects. Mathematical methods used to identify the presence or absence of path dependence, including an analysis of Markov chains in the attainment of successive "states" of project growth, may be employed in this analysis.

---

[17] J. Mateos-Garcia and W.E. Steinmueller, "The Open Source Way of Working: A New Paradigm for the Division of Labour in Software Development?" Falmer, UK, SPRU – Science and Technology Policy Research, INK Open Source Working Paper No. 1, January, 2003 [available at: http://siepr.stanford.edu/programs/OpenSoftware_David/The%20Open%20Source%20Way%20of%20Working.html]; J. Mateos-Garcia and W. E. Steinmueller, "Population Dynamics of Free & Open Source Software: A Research Roadmap," Presented at the OWLS Conference, Oxford Internet Institute, 25-26th June 2004 [available at: http://www.oii.ox.ac.uk/fiveowlsgohoot/postevent/Mateos-GarciaOWLS.pdf ].

[18] In contraposition to this tendency, there will be developers who are abandoning old projects as these reach their end, or because their interest waned. Individuals seeking to increase their status within the community may have incentives to terminate their roles as collaborators on existing projects in order to start new ones (a possibility will be considered later). If individuals derive utility from the excitement associated with "new hacks", persisting attachments to projects – of the sort described in the text -- would be less likely to be formed; indeed, were the typical exit/entry rates of developers participating in open source projects to remain high, that would suffice to mitigate the problems of secularly rising resource immobility within the community as a whole.

- ***Sponsorship support and relations between individual developers and commercial sponsors:***

This component of our research program is concerned with understanding the formation of the variety of complementary software products and services that commercial ventures have developed around the software-system code created by the larger "open source" projects. These activities are a source of direct support for some F/LOSS projects, and a beacon that may affect the launching of new programs, stimulate individuals to enter the community (which may result in their eventual participation in other projects that have no commercial ties), or signal which project is likely to achieve a critical mass of adopters. The degree to which such related, profit-oriented activities develop symbiotic connections with an open source project, rather than being essentially parasitic, can be investigated. But to do this would necessitate gathering evidence of individuals' overlapping participation in F/LOSS projects and commercial ventures based upon either proprietary or F/LOSS projects of both kinds; and by examining the formal commitments that are entered into in relation to existing projects.

A two-pronged approach has therefore been pursued to study the issues this raises.[19] First, a web-based survey of developers, the NSF-supported FLOSS-US Survey for 2003, has been conducted by the research group led by Paul David at Stanford University's Institute for Economic Policy Research. This survey replicated a number of the questions answered on the European Commission-sponsored FLOSS (2002) survey carried out under the leadership of Rishab Ghosh at the International Infonomics Institute at the University of Maastricht, but it elicited more detailed information from developers about their contacts with, and possible involvements in complementary/collateral commercial enterprises.[20] Where the two surveys overlap very similar patterns are observed in the responses, even though share western Europe residents among the FLOSS-US respondents is considerable smaller (c. xx percent) than that found among the FLOSS survey sample. Although there are some significant demographic and regional variations in the responses to the FLOSS-US, the following general picture emerges from a preliminary analysis of this data:[21]

a) F/LOSS developers tend to be highly educated and employed, with ambitions of future career advancement. Contradicting the impression of the open source community as being made up largely by students and otherwise unemployed "hackers," more than two thirds report themselves to be in paid employments. Regardless of whether they started writing "open source" code in the

[19] A third approach is under consideration: automated web-crawling searches to capture email addresses from proprietary software project sites and re-capture those addresses at open source software project sites may be feasible. Variations in an individual's email identities, however, would in all likelihood result in this method providing only lower-bound estimates of the extent of overlapping participation.

[20] The survey questions, and "The FLOSS-US Survey (2003) Report: A preliminary analysis" by P. A. David, A. H. Waterman and S. Arora may be obtained at: http://www.stanford.edu/group/floss-us/; simple tabulations of the results for each question are posted on the Web at: http://www.stanford.edu/group/floss-us/stats/. These are based on the 1,494 usable responses, out of a total of 1588 that had been received by the time the survey was closed (on 17 June 2003). Links to the FLOSS Survey Report (2002) are given at the former website. As well as increasing the sample density on particular questions, the replication of many of the questions has made it possible to establish the relationship between the two sample populations.

[21] For further details one should consult P. A. David, A. H. Waterman and S. Arora, "FLOSS-US: The Free/Libre/Open Source Survey for 2003," SIEPR-Nostra Project Working Paper, September 2003, [available at: http://www.stanford.edu/group/floss-us/report/FLOSS-US-Report.pdf].

1970's, 1980's or more recently, their mean age at the time was 26-27. The median of their starting ages, however, is closer to 22.

b) Contributing to the community of developers, promoting the F/LOSS movement, and improving software's functionality all figure frequently among the reasons that respondents list as having motivated them to become involved in F/LOSS development activities. Most respondents support the use of the GNU GPL and similar "open source" licenses as a means of protecting software users' general freedom, and ensuring that credit is given for their work. F/LOSS developers tend to believe their way of working can supplant much of proprietary software development.

c) Most developers report working on F/LOSS mainly on weekends and after the end of their employed workdays, although many work on F/LOSS in connection with their employment or studies. They spend the greatest amounts of this time coding, debugging, and testing software, rather than in other project activities (e.g., distribution support, administration, etc.).

d) As their careers in F/LOSS development progress, developers describe themselves as typically taking more influential roles in their projects; they also tend to work more hours per week and in more intense stints.

e) Approximately 50% of the respondents report having earned money directly or indirectly through their work on F/LOSS. Support for F/LOSS projects from external businesses and organizations has increased significantly since a decade ago, particularly since 2000. Over half of the survey sample population worked on F/LOSS projects that were being supported by external sources (including those being supported in higher education institutions).

f) Approximately 50% of developers launched their own projects, or were the "project maintainer" for their current project; the latter are typically small, and so correspond roughly with the "I-mode" (independent developer) type of project, rather than to the class of larger, "C-mode" (community) project organizations.

g) While most of the respondents report having contributed to only a few projects – a generalization that holds even when one excludes those who only recently became engaged in F/LOSS activities, there a small fraction (some 7-8 percent) of very active "core" developers who participated in many projects (the mean and median number of their projects being 5.5 and 6, respectively.). Approximately half of the developers say they wrote almost all of their most recent project's code, and an equal proportion rate their contribution to their current (and often their sole) project as "very important". Approximately one-third of developers say they contributed only incrementally to their most recent project.

By asking respondents to identify the first F/LOSS project on which they worked, and the two projects in which they deemed their participation to have been the most significant/important (for reasons given), the survey design has made it possible to link responses with the project characteristics information available from SourceForge and other open source project platforms, such as FreshMeat, and Savannah. The second related line of inquiry also connects with the work on the determinants of project "success", previously described: data available at SourceForge is being used to investigate whether there are significant statistical relationships between the specifics of the licensing arrangements adopted when projects are launched and the subsequent

development of derivative commercial enterprises around those projects that eventually do release code.[22]

- ***Using Simulation Modeling as an Integrative Device***

The fourth strand of the project is the development of a simulation model of the F/LOSS production system. This model-building activity aims eventually to provide more specific insights into the workings of F/LOSS communities. It seeks to articulate the interdependence between distinct sub-components of the resource allocation system, and to absorb and integrate empirical findings about the micro-level mobilization and allocation of individual developer efforts, both among projects and within them. Stochastic process representations of these interactions are a major tool in identifying critical structural relationships and parameters that affect the emergent properties of the macro system. Among the latter properties, the global performance of the F/LOSS mode in matching the functional distribution and characteristics of the software systems produced to meet the evolving needs of users in the economy at large is an issue that it is obviously important for our analysis to tackle in studying the questions this raises.[23]

To characterize the structure of the relative rewards associated with the adoption of various roles and participation in projects of different types, our initial approach has been to implement a sub-set of the "ethnographic" observations describing the norms of F/LOSS hacker/developer communities, notably following Eric S. Raymond's insights in the well-known essay "Homesteading the Noosphere".[25] The core of this is a variant of the collegiate reputational reward system: the more significance attached to a project, the agent's role, and the extent or critical importance of the contribution, the greater the anticipated "reward" in terms of peer regard, professional reputation and whatever psychological satisfactions and/or material benefits may be derived therefrom. Caricaturing Raymond's more nuanced discussion, we stipulate that launching a new project is as a rule more rewarding than contributing to an existing one, especially when several contributions have already been made; typically, early releases are more rewarding than later versions of project code; there are some rewarding "project-modules" that are systematically accorded more "importance" than others, and these are ordered in a way that reflects meso-level technical dependencies.

One way to express this is to posit that there is a hierarchy within a family of projects, such that contributing to, say, one or another of the many modules (or "packages") of code making up the Linux kernel is deemed to be a potentially more rewarding activity than providing a Linux implementation of an existing and widely used applications program; and,

---

[22] See J. Mateos-Garcia, "Factors in the success and failure of free and open source software projects: design of a study based on the SourceForge archives", June 2004. Available at: http://www.sussex.ac.uk/spru/publications/mateos_garcia.pdf.

[23] See J. M. Dalle and P.A. David, "The Allocation of Software Development Resources in 'Open Source' Production Mode", in *Making Sense of the Bazaar: Perspectives on Open Source and Free Software*, J. Feller, *et al*. (eds), Cambridge, MA, MIT Press, forthcoming in Winter, 2004.

[25] See Eric S. Raymond, *The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*, Sebastopol CA, O'Reilly, 2001, pp. 65-112.

correspondingly, a contribution to the latter would take precedence over writing an obscure Linux driver for a newly-marketed printer. In other words, we postulate that there is a lexicographic ordering of rewards following a discrete, technically-based "ladder" of project types. Lastly, new projects are always created in relation to existing ones, and here we consider that it is always possible to add a new module to an existing one, thereby adding new functionality, and we assume that this new module will be located one level higher up on the ladder.

As a consequence, all the modules of the project, taken together, are organized as in a tree which grows as new contributions are added, and which can grow in various ways depending on which part of it (upstream or downstream modules, notably) a developer selects. We further conjecture that the architecture of this notional "tree" will be to some extent correlated with both the project's actual directory tree and with the topology of technical interdependencies among the modules – although this correlation will probably be especially imperfect in the case of our initial specifications of the simulation model. A typical example of a simulated software tree is shown in Figure 1 below, where the numbers associated with each module represent versions, considering further that versioning accounts for to studying the issues this raises performance.

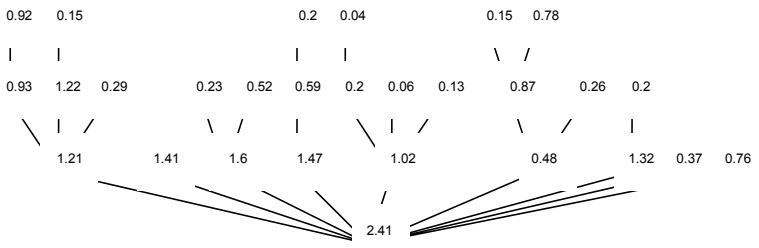


*Figure 1: A F/LOSS Simulation-Generated Software System*

With the help of such a simulation tool, we are then able to study social-utility measurements according to two basic ideas: (1) downstream modules are more valuable than upstream ones because of the range of applications that can eventually be built upon them, and (2) a greater diversity of functionalities (breadth of the tree at the upstream layers) is valuable because it provides software solutions to fit a wider array of user needs. In this regard, preliminary results indicate the social efficiency of developer community "norms" that bestow significantly greater reputational rewards for contributing and adding to the releases of downstream modules.

Further, these preliminary explorations of the model suggest that policies of releasing code *early* tend to generate tree shapes with higher social-efficiency scores. The intuitively plausible interpretation of this latter, interesting, finding is that early releases are especially important (adding larger increments to social utility) in the case of downstream modules, because they create *bases* for further applications development, and the reputational reward structure posited in the model encourages this "roundabout" (generic infrastructure) process of development by inducing individual efforts to share the recognition for contributing to downstream code.

The work described here is only a start on the integrative task of simulation modeling, and the agenda of work that lies ahead is consequently a long one. The behavior of developers (contributors) thus far is caricatured as myopic and, more seriously, it still lacks several important dynamic dimensions. Learned behaviors on the part of the developers, for instance, has not been allowed for in the pilot simulation model – a step that will make it necessary to keep track of the individual histories of the agents' participation activities. Acquiring the skills relative to a particular module is not without cost, and the model does not make any allowance for these "start-up" costs, which would also affect decisions to shift attention to a new package of code in the project. Further, and perhaps most obtrusively limiting, the first state of the model abstracts from heterogeneity in the behavior of the developers (in respects other than that arising from the endogenous formation of individual effort endowments); such differences could derive from the variety of motivations affecting the amount of effort that developers are willing to devote to the different modules, or to different projects. In particular, users are likely to have preferences for modules and projects that they will be able to use directly. To capture an effect of that kind will necessitate representing functional differences among software projects, and relating those characteristics to developers' "use-interests". We envisage such a simulation being employed to evaluate the influence of "user-innovators" – the importance of whose role in open source communities (as in other spheres of technological development) has been stressed in the work of Eric von Hippel.[26]

The personal rewards associated with contributing to the development of a project (whether psychological or material) will be most fully obtained only when the "maintainer" of the module or project accepts the code or "patches" that the individual in question has submitted. Rather than treating maintainers' decisions as following simple "gate-keeping" (and "bit-keeping") rules that are neutral in regard to the identities and characteristics of the individual contributors, it may be important to model the acceptance rate as variable and "discriminating" on the basis of the contributing individuals' "experience" or "track records". This would enable the model to capture some features of the process of "legitimate peripheral participation" through which developers are recruited. Contributions made to modules situated in the upper levels of the project "tree" (where comparatively fewer modules "call" them in relation to the modules on which they depend) might be supposed to require less developer experience and expertise for all significant likelihoods of being accepted by the project's maintainers. Comparative neophytes to the F/LOSS community ("newbies") would thus have incentives to start new modules or contribute to existing ones at those upper levels, but over time, with the accumulation of a track record of successful submissions, they would tend to migrate to work at lower branches of the trees.[27]

---

[26] See, e.g., E. von Hippel, "Horizontal innovation networks – by and for users," Massachusetts Institute of Technology, Sloan School of Management, Working Paper No. 4366-02, June 2002 [available at: http://web.mit.edu/evhippel/www/UserInnovNetworksMgtSci.pdf ]; K. Lakhani and E. von Hippel, "How open source software works: 'free' user-to-user assistance," *Massachusetts Institute of Technology, Sloan School of Management, Working Paper No. 4117*, May 2000 [available at: http://web.mit.edu/evhippel/www/opensource.PDF].

[27] The complex interplay of factors of learning and trust, and the ways in which they may shape the path-dependent career trajectories of members of the F/LOSS developer communities, have been carefully discussed in Juan Mateos-Garcia and W. Edward Steinmueller, "The Open Source Way of Working: A New Paradigm for the Division of Labour in Software Development?", INK Open Source Working Paper 2, SPRU University of Sussex,

All of the foregoing complicating features of resource allocation within and among F/LOSS development projects are more or less interdependent, and this short list is not exhaustive. There is therefore a great deal of challenging model-building work to be done, and further empirical research absolutely must be devoted to shedding light on these conjectures, and ultimately to permit approximate parameterizations of the most plausible versions of the simulation model. We believe that a modeling and simulation effort of this kind is a worthwhile undertaking because it can provide an integrative framework, assisting the synthesis and evaluation of the rapidly growing theoretical and empirical research literature on many distinct aspect of F/LOSS development. The results of the studies by "libre software engineers" and the findings of social scientists should be brought together to orient future model-building and -specification work, and should in turn be confronted with simulation findings obtained by exercising the model. It is important, too, that economic theorists become regularly engaged in dialog with empirical researchers, and so it is hoped that – uncommon as that seems to be in many fields of economics – the necessary forum and language for maintaining such exchanges will be provided by the availability of a flexible and "open" simulation structure. By pursuing this approach, it is to be hoped, it will prove possible eventually to bring social science research on the free and open source model of software development to bear, in a reliably informative way, upon issues of public and private policy for a sector of the global economy that is, manifestly, of rapidly increasing importance.

## III

### Envisaging the Future Trajectory of Research on Open source Software

To envisage the future trajectory of useful research concerned with open source software development, one has to begin by thinking about the likely trajectory of the phenomenon itself. To know where this dynamic process is heading, it helps to have a broader sense of "where it's coming from".[29]

The open source software movement is a "paradigm-shifting" development in the organizational history of the software industry. Software production has evolved through a succession of paradigmatic modes – originating in the vertical integration of hardware and software, achieving a measure of autonomy through the emergence of an industrial sector comprised of independent software vendors, gathering momentum through the general separation of hardware production from the software-systems development that marked the ascent of the mass-produced personal computer "platform". [30] The most recent production mode in the

---

2003. The abstract and text of this paper is available at: http://siepr.stanford.edu/programs/OpenSoftware_David/OSR01_abstract.html.

[29] The following paragraphs draw upon the statement presented by David and Steinmueller to the workshop convened at the National Science Foundation, Arlington, VA, 28 January 2002, on: "Advancing the Research Agenda on Open Source".

[30] W. Edward Steinmueller, "The U.S. Software Industry: An Analysis and Interpretive History", in David C. Mowery (ed.), *The International Computer Software Industry*, Oxford, Oxford University Press, 1996, pp. 15-52.

software industry, the open source mode, is closely connected with the continuing evolution of the personal computer platform into an information and communication appliance, a vehicle for network exchanges of digital data that is supplanting postal telecommunication for interpersonal communication and actively challenging the position of voice telephony.

"Communicating information appliances" must be enabled by software that provides for a far greater measure of technical compatibility and inter-operability than was the case when the dominant paradigm was the "stand-alone" personal computer. By the same token, the new paradigm has opened up the possibility of an entirely networked mode of creating, packaging, and distributing software systems – thereby marking a break with the era of "shrink-wrapped" software packages.

***Emerging challenges in the software industry***

In view of the disruptive character of the developments just described, it is not surprising that the dominant mode of economic organization and the dominant incumbent firms that had emerged from the personal-computer revolution are now finding themselves challenged. Indeed, quite a number of distinct challengers have emerged in the areas of software production, packaging and distribution – object-oriented mini-universes such as the JAVA world, the ongoing development of UNIX-based workstation environments, the new application platforms of mobile telephones with graphic displays, DVD-based games machines, and digital television. The distinguishing feature of the open source movement is that it is attempting to insert itself into the heretofore rigid link between a personal computer "desk-top" and the underlying operating system/application program environment, and, by so doing, to create an entirely different model for software acquisition, which would supplant packaged, shrink-wrapped software.

Open source software also is distinctive in discarding the industry's previously dominant business model: it contests the proprietary and quasi-proprietary systems based on the presupposition that software development costs must be recouped from the sale of individual "units" of software, integrated hardware and software systems, or service charges for using such integrated services (e.g. by playing digital-television games). This aspect of free and open source software is seen by some observers as defining the movement as a radical rejection of dependence upon the conventional intellectual property rights regime, and seeking to replace that mechanism for stimulating innovative activity with a voluntary "communal ethos" for the creation of intangible goods (in this case code) that possess the properties of "public goods".

Understandably, there is a good deal of skepticism about the realism of expecting the enthusiasm and energy that often attends the launching of collective undertakings to be sustained, and therefore to go on supporting and elaborating the highly durable artifacts to which their early efforts give rise.[31] For quite some time even sympathetic observers have been noticing that delivery has not been made on the promises that a new, viable business model would appear

[31] See, e.g., the strongly skeptical (but not entirely well informed) opinions expressed by a highly regarded legal scholar at the University of Chicago: Richard Epstein, "Why open source is unsustainable," *Financial Times—Comments and Analysis*, 21 October 2004 [available at: http://news.ft.com/cms/s/78d9812a-2386-11d9-aee5-00000e2511c8.html]. James Boyle's critique of Epstein's article the in FT.com ("Give me liberty and give me death?") is available from the same URL.

"real soon now", providing an alternative to the conventional business model based upon private appropriation of the benefits of invention and cultural creativity by means of legal intellectual property rights monopolies. Those who have been waiting for a new and economically viable *free-standing* business model for free and open source software, one uncoupled to any complementary commercial activity, may justifiably wonder whether they, too, are "waiting for Godot" But, instead any such miraculous business plan -- permitting the recouping of initial, fixed costs of open source code which is distributed at its marginal cost, along with all of the other elements of sunk costs associated with sustainable maintenance bug-tracking and patching activities, something else has emerged: the apparent willingness of profit-seeking producers of complementary goods and services to subsidize the production and distribution of free and open source software.

But, in addition to that somewhat surprising development, there are two potent forces that have continued to impart considerable momentum to the open source software movement. The first of these can best be described as a perfectionist impulse, charismatically projected by community leaders such as Richard Stallman, reinforcing the conviction that the evident virtues of voluntary co-operation will suffice to expand the availability of these software systems to the point where they will pose a full-scale challenge to the viability of the dominant commercial software firms. This can remain a potent force if it succeeds in bringing new members into the movement. The other driver of the movement is more of market "pull-force" than a social "push-force": the practical, purely instrumental need for a robust and "open standard" software environment to support the continuous availability of networked information resources. The attractive technological goal, then, is to fill the vacuum that has been left by the absence of any apparent winner among the available commercial offerings in the movement of local area network software products to the Internet and World Wide Web environment.

***The dynamics of "open source" as a socio-political movement***

The former of these two drivers may well succumb eventually to the dynamics typical of other charismatic movements: having thrust a few leading developers into international prominence, their followers gradually allow their own energy and attention to dissipate. The status gap separating leaders from followers widens, and low odds of replicating the spectacular reputational triumphs of members of the movement's vanguard slowly become more apparent (both factors taxing the abilities of even the most charismatic figures to animate the community as a whole); the day-to-day requirements of ordinary life, and the exigencies of earning a living overtake the idealist commitment that motivated many among the multitude of obscure followers. Yet, it is on precisely that (now flagging) enthusiasm that the demonstrated efficacy of this mode of production depended, and the load shed by the many among the disaffected must therefore fall more heavily, and eventually intolerably, upon the shoulders of the few who remain most committed. This is the skeptical, pessimistic sociological scenario depicting the fate of idealistic communalism in economic affairs. [32]

[32] It is quite possible, however, that before dissipating the movement would have become strong enough and sufficiently sustained to thoroughly disrupt and even displace the once dominant proprietary sector of the software industry. Consequently, the survival of the conventional commercial model is not implied by a prediction of the *eventual* decay of the free and open source software movement. History is usually messy. From the envisaged "crisis" that would overtake the commercial software industry, one socio-political solution that could well emerge might be something akin to the Tennessee Valley Authority program of rural electrification -- massive public

The optimistic scenario, on the other hand, highlights the possibility that the potency of the demand-pull force may well survive to become a sustaining factor, because in essence it involves a re-integration and viable reconfiguration of numerous constituencies that share an interest in information technology and that also possess the skills and have the personal commitments necessary to impel them to continue working that field. A division of labor between the large population of individuals who have secured employment in managing the new infrastructure of what is referred to (in Europe) as "the Information Society", and the hardware companies that are responsible for the physical artifacts of that infrastructure, may suffice to maintain this mode of software production. But it would have to do so in an environment in which access to the Internet markedly lowers the costs and organizational challenges of promoting and distributing innovative software solutions.

To date, such social science research attention as has been devoted to the open source software movement has focused (understandably) upon the role of leaders, and the interpretation of the variety of tracts emanating from the most charismatic branch of the movement. This line of inquiry is topical, as well as intellectually engaging: it provides the basis for understanding the conditions on which individuals are recruited into the movement, and how their interest and commitment are maintained throughout the arduous process of creating useful and reliable software. It also provides an observational field for the systematic re-examination of the sociology and economics of voluntary association, the organizational processes governing the definition of goals and the achievement of "closure" in reaching goals, and the formation and functioning of an interface between open source and commercial efforts. In particular, it offers an illuminating set of comparisons with the governance norms and organizational structures of co-operation found in other universalistic, distributed, epistemic communities that have created their own, collegiate, reputational reward systems – such as the research communities working in academic "open-science" institutions. The latter, similarly, exist only through patronage, or the formation of other symbiotic relationships with agencies that furnish the participants with material support for their creative endeavors.

All of the immediately foregoing lines of inquiry are, in one way or another, and in varying degree, threaded through the agendas of social science research projects already underway throughout the world, including the one we have described in the preceding pages. It should be rewarding, and it may be possible, to venture still farther afield by beginning to think about research agendas that would direct attention to the second "branch" or "force" that may well continue to sustain the growth of the open source software movement. That, at least, is our purpose in the following paragraphs.

***Implications of prospective advances in information technology***

It is important, first, to consider the significance of the next important development in information exchange standards, XML, which transcends the powerful but limited capabilities of the HTML standard that (with extensions) has to date been driving World Wide Web developments. XML provides a much broader base for creating complex informational artifacts and, correspondingly, has an enhanced capacity for the development of proprietary tools to exploit these capabilities. In the later stages of HTML, technical compatibility issues in

---

subsidization and regulation of the production and distribution software, seen as the critical technological infrastructure for the knowledge economy.

maintaining web sites have favored the growth of proprietary tools – i.e., sites are increasingly created and maintained using a single-platform design package. This development can be empirically detected by the automated methods emerging from the Internet research field.

How the community responsible for creating and maintaining the information infrastructure will respond to this development is not at all clear. On the one hand, its members may refuse to be tied to proprietary platforms for content creation because of the inevitable cost of such systems. If such is the case, a focus for future open source activity may become the building of the tools used to create HTML/XML content. On the other hand, that community may embrace commercial packages, creating a major division within the open source community between those concerned with "lower layer" connectivity and those concerned with "higher level" content.

Second, the relationship between open source software and peer-to-peer networking movements warrants closer scrutiny. On the one hand, peer-to-peer has been a major instrument in what is described in some circles as "direct action against usurious copyright fees", and in other circles as "large-scale piracy". On the other hand, peer-to-peer extends the Internet's function as a publishing engine, thereby providing the basis for a new exchange economy. Systematic research into the nature of the assets being created and exchanged within this economy, and the response of the developer communities involved to sustained efforts to suppress the abridgment of intellectual property rights, would provide an early indicator of the new patterns of information production and exchange that are likely to emerge towards the end of this decade and during the next one.

Third, a basic characteristic of open source software communities that has also been undergoing development and elaboration in other contexts, such as computer gaming, is the systematic and explicit assignment of "status" to community participants by their peers. Systems such as that developed by Advogato involve interesting voting rules and procedures for determining user valuation, and these are worth analyzing in the light of the theoretical and empirical social science literature on "demand revelation" and "preference aggregation" mechanisms. Further research may well need to be focused on the technical and social factors involved in deliberately constructing peer-based "status systems," including the creation of a capacity for codified, formalized and automatically generated reputational hierarchies to motivate and direct the efforts of individual participants, and mechanisms for reducing the "voting costs" of generating the information that such systems require. Research findings in this area could serve a variety of practitioner and policy communities alike, by indicating how best to create complex goods under conditions of asymmetric information and high monitoring costs.

Fourth, the interface between open source-type distribution and other forms of publication and distribution deserves greater attention. A variety of new intermediaries have emerged in the industry publishing e-books, music, and other information commodities. Some of these are operating within a full commercial model, while others (such as the long-established Xanadu project) utilize a variety of public information models. The relative performance of these different communities in achieving goals of distribution and access provides important information about the long-term viability of public information creation and distribution systems, as well as quasi-public good-production modes such as clubs and voluntary consortia.

Consider how best to proceed along the last-mentioned line inquiry immediately raises the more difficult and longer-term challenge of envisaging the future structure of processes of information creation and exchange, and the problems of devising incentive systems that will be compatible with the future production and distribution of information, including scientific information. The sense of enjoyment derived from being attached to (embedded in) a community engaged in some higher, trans-individual (tribal?) purpose is a source of satisfaction that many reasonably well-paid professionals seemingly find hard to obtain in their work as members of hierarchically managed profit-seeking organizations. The Internet would appear to have addressed that need, at least in part. On the evidence of both the tenor of survey responses from a substantial proportion of the developer community, and the impressively complex and reliable software products created by the large open source projects, the formation and support of "virtual communities" of co-operating software developers serve to mobilize participants and satisfy their (otherwise unfulfilled) need to enjoy the exercise of their skills. Moreover, the Internet allows them to enjoy the exercise that skill at convenient times and at pecuniary costs that are low in comparison to those entailed by other, more conventional modes of production.

In this regard, the Internet obviously has great advantages of size and speed of communication over the means that enabled the formation of networks of correspondence and association among the amateur gentlemen scientists of the early 19th century. Is the voluntaristic impulse to create and share knowledge – now manifesting itself in a great variety of virtual communities on the Internet, one of them being the open source community – likely to increase in scale and scope with the growth of real income and the liberation of larger proportions of the world's population from physical work? This is a question that economists can usefully tackle, even if certainty in prediction remains elusive. At the very least, it appears that they may in that way make considerable progress towards identifying the "boundary conditions" within which such voluntary productive entities can expand and be maintained. Other social- and behavioral-science disciplines may then be left to seek the sources of individual psychological motives and social cohesion that occasion the emergence of such movements and energize their members.

## IV

## Summing Up

Every one of the subjects that have here been identified as warranting further investigation takes as its point of departure the existence of a community (usually a virtual community) striving to assemble the tools and organizational resources necessary to accomplish some purpose. The open source mode of software development may constitute a paradigmatic framework for collective creative activities whose scope extends far beyond the writing and debugging of computer code. To develop the means of assessing how, where, and why this and other related frameworks succeed in supporting other specific objectives – and where they are likely to fail – is both a challenge and an opportunity to contribute significantly the advancement of the social sciences, but also, and even more significantly, to effective human social organization. Indeed, in this exciting and important research area, there is ample work to engage for many hands and many minds.