Open Sourcing Nanotechnology Research and Development: Issues and Opportunities

Bryan Bruns
Sociologist, Independent Consultant

Prepared for the Foresight Conference on Molecular Nanotechnology, Bethesda, Maryland, USA, November 3-5, 2000

Working Draft 0.8 October 29, 2000 [available via www.BryanBruns.com]
Please send comments to: BryanBruns@BryanBruns.com

Abstract: The prominent role of software in nanotechnology research and development suggests that open source development methods might offer advantages in improving reliability, performance and accessibility. Open source approaches have shown new opportunities for voluntary cooperation to create and improve complex software. Suitable software licenses could be used to promote access, compatibility and sharing of improvements. Many companies currently associated with nanotechnology produce materials, equipment, and research and development services, all of which could support open source business models; however no company yet emphasizes an open source strategy. Some molecular modeling software is already open source or public domain. Software for molecular engineering constitutes an important opportunity for open sourcing, especially if systems architectures encouraging collaboration can be further developed. Analysis suggests that the net impact of open sourcing would be to enhance safety. Initiatives for open sourcing of molecular nanotechnology could be strengthened by coalition building, and appropriate strategies for open source licensing of copyrights and patents.

Contents

Open Sourcing Nanotechnology Research and Development: Issues and Opportunities

BryanBruns@BryanBruns.com

Summary

Opportunities. This paper analyzes the applicability of open source models for software development (Raymond 1999) to molecular nanotechnology (Drexler 1986). Open source approaches could offer several potential advantages for nanotechnology research and development. Broad peer review to detect and correct bugs and make improvements contributes to reliability and performance features. Open source is customizable, avoids dependence on a monopoly supplier and lowers costs. Open source efforts rely on voluntary network cooperation facilitated by the low cost of duplicating and communicating computer based information. Collective action to create intellectual common property offers an alternative which may both complement and substitute for proprietary development of closed intellectual property. Open source methods can help help avoid the costs (rents) of monopoly intellectual property and reduce the risk of anti-commons failures which may be created by the difficulty of negotiating rights. Open source methods have demonstrated the potential to promote rapid and accessible software development.

Licensing. Open source software licenses offer useful precedents for MNT. Desirable license properties would include the core Open Source characteristics enabling royalty-free redistribution and modification. Many of those concerned with open source licensing favor requiring disclosure of derivative modifications, and use of modular interfaces. The prominence of the GNU General Public License in applied and scientific software makes it desirable that licensing be compatible with the GPL. The role of hardware, and of businesses, favors licenses which enable combination of open intellectual property with closed software and hardware. This GNU Lesser General Public License is an appropriate license for those who wish to require disclosure of modifications while still permitting creation of combined works with closed source software. However, simpler licenses such as the X or modified BSD license can enable sharing of source code, while avoiding much of the complexity and confusion of the more restrictive licenses. Open sourcing of hardware patents needs further attention. Patent pools enabling access to open sourced intellectual property rights could offer an additional inducement to contribute to open source efforts.

Business models. Open source business models could be feasible for many of the companies currently associated with nanotechnology, given their reliance on sales of hardware and other goods, and customized services such as contract research and consulting. No "nano" company yet makes open source a major part of its business strategy, although there is a pragmatic willingness to use and support open source software. Business models compatible with open source offer advantages including reduced risk and greater robustness in dynamic environments. Consortia for Open MNT efforts might be supported by companies which want to ensure access to nanotech as a production technology. Open source approaches might be particularly useful where consortia wish to avoid antitrust problems.

Software. The significant amount of molecular modeling software that is already open source or public domain offers a useful demonstration of the relevance of open source, and a basis for further work. Several efforts have been made to develop molecular design software specifically intended for MNT, but these need to attract a larger pool of potential contributors before they can flourish. The ways in which systems architecture evolves could have major implications for whether MNT software and hardware systems become monolithic and proprietary, or develop as open architectures with modularity and levels encouraging competition and combinations of open source and closed source components. The best prospects for open source may lie in infrastructure-type standards for basic services, safety assurance and customizable design of nanoproducts.

Safety. Safety in molecular nanotechnology is mostly a long term issue, but need to be addressed since discussion of open source may evoke concerns about risks. Open source methods could offer substantial advantages for reliability and safety in normal applications of MNT. Analysis of threats from weapons based on nanotechnology already assumes that the technology could not be kept secret, so these concerns would be largely independent of whether open or closed source is used. Open source could contribute to the success of efforts, such as those suggested by the Foresight MNT Guidelines, to design safety into the architecture and elements of nanotechnology, and might similarly contribute to other methods for preventing or defending against dangers. Net impacts of open source are likely to be positive in the long term, and would seem to offer significant advantages in the short and medium term.

Conclusions. Open source approaches offer a method feasible under current intellectual property systems for promoting cooperation to develop better and more accessible molecular nanotechnology. Suitable licenses are available for software, while hardware licensing and the potential for patent pools deserve further exploration. Companies interested in nanotechnology could consider pure or mixed strategies for using open source methods, and possible formation of consortia. In the near term, software for molecular design offers the most promising pathway for open source initiatives in molecular nanotechnology. The application of open source methods to research and development in molecular nanotechnology deserves further exploration.

Introduction

This paper analyzes the applicability of open source models for software development (Raymond 1999) to molecular nanotechnology (Drexler 1986).[1] The rise of the Linux operating system has brought increasing attention to the potential of open source software development. Open source approaches have revealed new alternatives for managing large-scale software projects (Brooks [1975] 1995), coordinating the creation of easily accessible software tools and escaping domination by corporate software monopolies. In the longer term, nanotechnology offers the promise of "making matter like software" (Ellenbogen 1997), so that software characteristics of low cost reproduction and dissemination would apply to material goods. Open source approaches could greatly expand access to future benefits from molecular nanotechnology. In the nearer term, which is the focus of this paper, open source approaches might offer advantages for faster, more reliable and more accessible research and development.

Organization of the paper. The paper begins by identifying potential advantages from applying open source approaches to creation of shared intellectual property. Open source principles are defined in the licenses which shape access and incentives. Looking at companies currently associated with nanotechnology helps assess the potential feasibility of open source business models. The current state of molecular modeling software reveals opportunities and constraints on the growth of open source approaches. Safety implications are a potential source of concern, but further examination suggests advantages of open source approaches for safety. The paper concludes that open source software offers substantial advantages, which deserve further exploration. Readers familiar with open source, and more interested in practical implications than institutional design and licensing rules, may wish to first read the sections on business models, software, and conclusions.

Opportunities

Ideas and eyeballs. Open source methods enable widespread, low cost access to software, opening it for sharing in an "information commons." In his influential essay, *The Cathedral and the Bazaar*,[2] Eric Raymond argued that the principal advantages of open source approaches lie in producing better software by mobilizing a large community of developers and users who are able to examine open source computer code to detect and correct errors (bugs) and improve features. More eyeballs find more bugs and more ideas yield more solutions. These advantages may be particularly strong where errors are hard for the original programmers to find, so that peer review and widespread testing can yield major benefits for reliability. Raymond argues that reputation provides the principal incentive for the programmers contributing to open source efforts in "gift cultures," though the inherent satisfactions of craftwork and other motives also have a role.

Economics of open source. In *The Magic Cauldron*, Raymond notes that most software is written for use rather than sale, which helps explain the economic rationale for open source software. In most cases, programmers and their employers want software that helps them produce other goods and services, rather than hoping to profit from selling the software itself.

They use and contribute to open source software because it enables them to:

- customize the software for their own needs,
- correct faults without waiting for someone else,
- reduce the risks of software obsolescence,
- avoid dependence on a single monopoly supplier and
- reduce costs.

Organizations may support open source efforts based on these advantages, and as part of the business models discussed later in the paper. All the advantages of open source seem potentially applicable to nanotechnology, both for software used in current nanoscience activities such as molecular modeling, and the software and equipment which might be employed for design and manufacture of future nanotechnology tools and products.

Avoiding anti-commons. Concerns have been expressed about how current intellectual property rights systems might hamper development of molecular nanotechnology (Peterson 1999, Drexler 1999, Krummenacker 2000).[3] The practical worry seems to be that problems with intellectual property rights, particularly patents, might impede the ability to use ideas in research and commercial applications, hindering the development of MNT. In a more general discussion, concerned mainly with biotechnology, Heller and Eisenberg (1998) suggest that excessively fragmented intellectual property claims could result in an anti-commons. Knowledge, enclosed by exclusive intellectual property rights, might go unused because of the transactions costs of negotiating the necessary agreements among a multitude of owners, with divergent interests, and incompatible expectations about the values of their intellectual property. Speculative pursuit of rents from intellectual property could cripple the growth of knowledge. Patents inherently result in underuse, as a result of raising prices and restricting use, but in theory this may be offset by advantages of encouraging innovation and requiring disclosure of ideas. Heller and Eisenberg argue that current tendencies toward excessive and inappropriate enclosure of intellectual property could be counterproductive, using examples from biotechnology. They discuss the possibility of collective action and institutional innovation to share patents, noting examples such as the complex arrangements for sharing income used by the music industry, but suggest that antitrust fears and other problems might deter such innovation. Doll (1998) and other commentators suggest that fears about deterring subsequent innovation and commercialization may be exaggerated or unfounded. Open source licensing arrangements are not discussed by Heller and Eisenberg, but offer an alternative route to avoiding anti-commons problems by encouraging sharing of intellectual property.

The logic of information goods. Cooperation to create information goods constitutes a problem of underprovision (Raymond 1999), rather than the problems of depletion and congestion often discussed in terms of the tragedy of the (unmanaged) commons (Hardin 1968, 1998; Ostrom 1990). Solving "free-rider problems" (Olson 1971) requires forming a coalition sufficient to provide the good, overcoming incentives which might motivate users to benefit from the good without contributing to its provision. Olson suggested that cooperation was easier to negotiate and monitor in "small" groups, but that larger groups required selective incentives exclusively available to contributors. Open source projects are typically initiated by a participant with a large stake in their success, characteristic of what Olson called "privileged groups." Lower transactions costs for communication and duplication of information facilitate cooperation among widely-dispersed participants, dramatically expanding the envelope of opportunities for open source efforts. Pursuit of private reputation, and use of licenses to regulate access, provide selective incentives to encourage contributors and reduce risks of exploitation by free-riders. As discussed later in the paper, further opportunities for strengthening selective incentives in large-scale open source efforts may exist through such means as formal consortia and pooling of intellectual property rights.

Limits of reform in proprietary intellectual property. Options for open sourcing nanotechnology need to be considered in the context of other possible changes in intellectual property rights. This paper assumes that during the next few decades, development of molecular nanotechnology will likely take place under intellectual property regimes similar to those which prevail at the present. Current systems might be refined by such measures as improving the institutions which issue patents, particularly by increasing access to information and opportunities for public comment about the "prior art" of what has already been patented or put into the public domain (see for example the Software Patent Institute (2000) Database of Software Technologies). More substantial changes may come through new laws, court rulings and broader social and technological changes. These may affect the scope of what can be patented or copyrighted, the duration of rights, the penalties for violations and the ease or difficulty with which rights can be enforced. Reform efforts in the US might target reform of the policies which currently strongly promote private appropriation of inventions created as part of government-funded research and development. Nevertheless, patents for industrial processes and equipment, as well as copyright protection for software, seem likely to persist.

The intellectual property system has resiliently survived and evolved through radical changes in science, technology and society over the past few centuries. There are strong commercial interests in favor of the current system, as well as popular legitimacy and articulate supporters of the rationales for intellectual property rights. The complexity of intellectual property law, and increasing linkages with international treaties such as the World Trade Organization, further reinforce the inertia of the current regime. Proponents of MNT and others concerned about it may well devote some effort to lobbying for reforms in intellectual property rights. However, given the likely limits on major reform in the near term, and the rapid advance of current nanoscience, it seems worth considering strategies, such as open sourcing, which can be implemented without drastic changes in the current intellectual property rights system.

What not to open source. The arguments for open source do not imply that all information should be placed in the commons. Open and closed source intellectual property each have advantages under different circumstances and can productively coexist (Raymond 1999:169-179). The arguments against open sourcing are strongest for valuable secrets which could generate large profits, either from royalties or by blocking competitors, therefore companies would not want to give away their "crown jewels." However most software is not such a source of competitive advantage. So it is worth considering whether the benefits from using open source tools outweight any potential gains from secrecy or being able to deny competitors access to particular knowledge and techniques, based on proprietary intellectual property guarded by copyright, patents and trade secrets. Raymond suggests that open source may be most relevant for more basic infrastructure-type standards and for implementation of widely known techniques, while being less useful for innovative applications. Later sections of the paper discuss some of the factors which could influence decisions to use open source approaches. The option of open source should at least be considered, rather than assuming that closed intellectual property is the only way to go.

Creating information commons. The creation and defense of proprietary intellectual property entails substantial transactions costs. It risks "anti-commons" problems by impeding the use and refinement of ideas. Easy duplication and communication with computers and telecommunications now facilitate voluntary network cooperation for the provision of shared information goods. This occurs in an "inverse commons" where having more users tends to increase, not decrease, the value of commons in networked information (Raymond 1999). Low transactions costs and reputation-based "gift cultures" facilitate open source as an alternative or complement to the more familiar organizational modes of company hierarchies or market contracts (Coase [1937] 1988, Williamson 1996, Bennett 2000). In some cases, open source arrangements for creating useful knowledge may have lower overall transactions costs for society (Bruns 2000). Incentives for contributing to the provision of information commons are reinforced when ideas are not just put into the public domain, not simply "open access" free for the taking, but instead subject to licensing rules.

Licensing

Licenses permit others to use and improve open source software. The licensing arrangements developed for open source software offer useful examples of ways to encourage sharing of information in nanotechnology research and development. While open source software discussions have mostly focused on copyrights, the role of equipment in nanotechnology requires additional attention to patents for hardware.

Defining Open Source. Open source approaches incorporate a range of principles for ensuring availability and freedom to use, modify and redistribute software. The Open Source Definition (2000) specifies the following characteristics:

- Free redistribution must be allowed
- The source code must be available
- Modifications and derived works must be allowed, and distributable under the same licensing terms as the original software
- No discrimination against persons, groups or fields of endeavor
- Licensing not limited to a particular product or particular software distribution

- No restriction on other software distributed along with (but not combined with) the licensed software

Open source licenses. Software placed in the public domain can be used by anyone, however they wish, including reissuing the same software, or a modified form, with a commercial license. Software protected by copyright and patent can only be modified or redistributed as explicitly permitted by the owner. Typically the source code is not disclosed, only compiled binary versions are distributed for execution. By contrast, open source licenses, relying on copyright, patent and other intellectual property law, provide rules to make code available for others to use, modify and redistribute in accordance with specific principles.[4] This can promote confidence that the open source software will continue to be available and be improved, and encourage contributors to help improve the software. Aside from the rights assigned by a license, the original rights holder may still retain privileged rights to issue other versions or modifications under different licenses, for example on another platform or for customized versions. While there is substantial consensus about the principles outlined above, there are variations among some of the major licenses, as Table 1 shows, concerning whether the source code for modified versions must be disclosed and whether modifications may be combined with closed source code into larger combined works.[5]

- Licenses, such as the revised Berkeley Standard Distribution (BSD), X11 project, MIT, and Python licenses require acknowledging the originator, but otherwise do not restrict further use. All or parts of the code may be reused in closed-source commercial products.

- The Mozilla Public License (MPL) and many recent licenses with similar principles require that the source code for any modifications be made available, as a way to encourage sharing of improvements within an open source community. However original or modified source code covered by the MPL may be combined in larger works with separate files which have no covered code and are not disclosed.

- Similarly under the GNU Lesser General Public License (LGPL), source code for modifications must be shared under the LGPL. However modules covered by the LGPL can be linked with closed source software in combined works without requiring that the entire work be under the LGPL.

- The Gnu General Public License (GPL) is one of the oldest and best known licenses, which pioneered the concept of using a license to deliberately promote the development of "Free" (as in free speech and liberty) software. Its "copy-left" design deliberately requires that any program which incorporates GPL code must also be under the GPL. This prevents code which has been shared from being "taken private" again in closed commercial products, and is intended to promote a growing body of "Free Software." The creation of "Free" versions of Unix, including GNU/Linux is the most dramatic example of software developed under the GPL.

Table 1. Comparison of open source licenses

	a. Public Domain	b. BSD, X11, MIT, Python	c. MPL and LGPL	d. GPL
1. can be re-licensed by anyone on any terms	yes	no	no	no
2. can distribute derived works without disclosing modifications	yes	yes	no	no
3. can incorporate in a combined work with closed files or modules	yes	yes	yes	no

Sources; Adapted from Perens 1999, Open Source Initiative 2000 and Hecker 2000
Licenses
b. BSD=Modified Berkeley Standard Distribution; similar provisions apply for licenses including, Python, X11,=X.Org/Open Group;and MIT=Massachusetts Institute of Technology;
c. MPL=Mozilla Public License (Disclosure requirements are also part of the Artistic License, IBM,License for the Jikes Compiler, Ricoh and other licenses;
LGPL=GNU Lesser General Public License;
d. GPL =GNU General Public License

Multiple-licensing and hybrid strategies. Programs under the X, MIT, BSD and Python[6] licenses can be incorporated into works issued under other licenses such as the LGPL and GPL with no problems. The MPL and IBM licenses include restrictions which make them currently incompatible with the GPL. Issuing under multiple licenses is however feasible, as in the Mitre Public License for their Collaborative Virtual Workspace, which allows use under either the GPL or MPL. Netscape has recently announced its intention to dual-license the Mozilla browser, permitting use under both the GPL and MPL licenses, and possibly also under the LGPL. Hybrid licensing models (Raymond 1999:168, Hecker 2000) are also possible, for example requiring royalties under a commercial license during an initial period before later release as open source. Examples of hybrid strategies planned in advance seem uncommon so far, but decisions to shift to open source after initial release under closed commercial licenses result in de facto hybrid strategies.

Open source licensing for patents. Open source software may include licensing of rights to royalty-free use of relevant software patents. Controversy about the legitimacy of software patents has been accompanied by practical approaches to keep software patents from interfering with sharing of software. The Gnu General Public License requires allowing use of patents as part of unlimited royalty free redistribution (except for possible geographic restrictions): "we have made it clear that any patent must be licensed for everyone's free use or not licensed at all." The Mozilla Public License seeks to restrict the patent license to use as part of the original or modified code, disallowing any separate use. Both the MPL and IBM's public license include stipulations that anyone who makes legal challenges to the relevant patent rights of the originator or contributors forfeits their access under the license.

Open Hardware Licenses. Various groups are working to develop "open" computer hardware including motherboards and other equipment, as well as associated interfaces, drivers, designs and tools. One of the leading figures in formulating the Open Source Definition, Bruce Perens, started an Open Hardware initiative aimed at printers and other hardware needing software drivers. A variety of licenses are being used or developed in the open hardware projects.[7] Reviewing several of these shows a pattern of requiring disclosure of modifications. The licenses show flexibility about the extent of use with non-GPL software and proprietary equipment. The issue of combinations between open and closed software led to the creation of the LGPL. Similar combinations seem inevitable for a hardware projects, given the current predominance of proprietary hardware, though several of these projects aspire to change the situation in the longer run. Observation of these hardware efforts suggests that, for nanotechnology related initiatives, purist insistence as done in the GPL on use only with other software and hardware complying with a highly restrictive license is unlikely to be workable for hardware. However if the views of those taking part resembled participants in the open computer hardware initiatives, then they might insist on licenses which require disclosure of modifications, rather than the less restrictive approach of the X11/BSD type licenses.

Patent pools. The Open Patent project (Shewmaker 2000) seeks to develop a general license which could be used for software patents. The current goal for the Open Patent project seems focused on supporting Open Source software and resolving problems resulting from software patents. Participants would join patent pools created by cross-licensing. These would be designed to create leverage by offering access to successively larger pools of patents. The key incentive for participants would lie in access to a pool of patents, avoiding costs of negotiating access and paying royalties. Like the arguments for open source above, this draws on the conceptual shift to emphasize a company's role as a user of intellectual property, emphasizing the interest it has in benefiting from easier access. The current draft proposals for the Open Patent are relatively complex and ambitious, with several different pools intended to induce sharing of all types of intellectual property held by a company, not only the specific items to which rights would be granted in a particular piece of software. The core concept of developing an attractive patent pool would seem to have substantial merit for opening access to software and hardware patents, even if it started from a narrower MPL or LGPL-type license. As with requirements for disclosing modifications in software, creation of a broader patent pools might be considered a sufficiently attractive inducement to merit special attention in designing new or modified open source licenses for MNT.

Publication and antipatents. Establishing patents is a costly and uncertain process, much more difficult than establishing copyright. Publishing new knowledge into the public domain avoids the costs of establishing patent rights. Licensing mechanisms for sharing intellectual property would be most relevant for those cases where patent rights had already been established, even if only for defensive use or as assets to aid in negotiating cross-licensing agreements. For those who do not seek patent rights, defensively publishing a potentially patentable invention puts the idea into the public domain, and so, in theory, should make it unpatentable. IBM is reported to have issued Technical Data Bulletins for ideas which it did not seek to patent, but wanted to be sure no one else would patent.[8] Rebecca Hargrave and Carl Malamud (2000) argue in their essay, Transparent Patents, for improving availability of "prior art" and review of patent applications. They also recommend arrangements for publishing "antipatents" which disclose enough information to prevent subsequent patenting.

Advantages of the LGPL. A license suitable for use in MNT would need to permit combination with proprietary software and hardware. The LGPL is somewhat more specific than the MPL or other restrictive licenses about the requirements for incorporation in combined works. However modular design and interaction, for example using application programming interfaces (APIs), is likely to be part of the architecture of software and hardware in nanotechnology, so the requirements in the LGPL should not create a significant obstacle. Further refinements regarding patents and patent pools might be developed. For cases where patents are obtained, for whatever reason, release under a simple license is an option worth considering for those who hold full rights to their software and are not concerned to restrict reuse. It minimizes restrictions, confusion or hassle for those who may wish to reuse the software under the MPL, LGPL, GPL or other stricter license. For those who want to insist that modifications are shared, or who are building on software which already incorporates such disclosure requirements, then the LGPL seems to offer an appropriate and useful license for promoting open source efforts in nanotechnology. The main drawback of the LGPL is that its use is explicitly discouraged ("deprecated") by the the Free Software Foundation, based on their objective of establishing a separate body of "Free Software." This raises questions about future support for the license and how conflicts regarding the interpretation of the license might be handled. Ideally it might be useful to have a license which requires disclosure of modifications and is under the control of a supportive, non-commercial organization. In practice, the risks of confusion and conflict regarding the LGPL and MPL add weight to the arguments for using simpler, less-restrictive licenses such as X11 and BSD, which are also still compatible with the GPL.

Hardware patent licensing. The key role of physical equipment in nanotechnology means that patents are likely to play an important role, so open sourcing of nanotechnology would need to include arrangements for use of patents. This could be either on a fully royalty-free basis, or compromise arrangements involving patent pools or other arrangements to prevent anti-commons problems and promote accessible software. In the longer run, licenses need to be developed, or borrowed from other sources, which enable suitable release of patent rights not just for software but for physical devices and processes.

Business Models

Business models. In his essay *The Magic Cauldron,* Eric Raymond (1999) identifies several business models through which companies can profit from open source software by:

- Reducing costs and risks through joint efforts in developing open source programs.
- Providing specialized services such as preconfigured installation packages, technical support and customization.

- Stimulating sales of hardware.
- Selling accessories including manuals, training courses, certification, franchises and other complementary goods and services that enhance the software.
- Building market share and name recognition, and
- Enabling access to content, such as news or specialized databases.

Applying the models. Most of these business models seem directly relevant to nanotechnology. Companies could rely on a core competence in applying open source tools to specific applications. Nanotechnology will need specialized equipment and materials. Manuals, textbooks, training courses and other accessories are likely to be increasingly profitable as interest in the technology rises, offering sources of revenue for companies and experts with skills in nanotechnology. Availability of equipment capable of building nanodevices would create markets for specialized designs. All these models could encourage businesses to help finance open source development and be relevant for startup or existing businesses interested in nanotechnology.

Mixed strategies. The feasibility of open source business models does not mean that all companies could or should pursue open source strategies. There may well be many cases where companies would want to hold onto their "crown jewels," intellectual property which enables them to gain a major competitive advantage, or extract royalties from commercial licensing. Use of open source may be selective. The key point is that a business model relying only on proprietary intellectual property is not the only choice. Open source alternatives deserve consideration, especially where open source helps a company to improve its core competence and better cope with change.

Producer coalitions. Potentially the most significant opportunities for open source approaches come from the role of open source software as an input. This draws on the key insight that most software is written for use, not sale. This illuminates a host of opportunities for open source. Companies which want to use nanotechnology to enhance their business could have strong incentives to support a coalition to develop open source nanotechnology. Rather than "betting the company" on being the first to commercialize some proprietary form of nanotechnology, companies could reduce costs, reduce risk and enhance their survival prospects by supporting joint open source efforts. This might be a very attractive strategy for large existing companies. A nanotech company willing to commit to such a strategy might be able to attract investment for itself or for efforts funded through a consortium. Such arrangements already exist in the information technology industry, which is already driving substantial investment into nanoscale engineering. A narrow consortium might develop primarily for members' use, offering access as an incentive to investors. A wider consortium backed by major players might choose to attract broader support, enhance legitimacy and avoid potential antitrust complications by using fully open source approaches.

Current industry structure. Industrial efforts for molecular nanotechnology are only in the earliest phases of gestation and the only certainty is that surprises will occur. Examining companies listed on two nanotechnology websites provides one way to take a brief look at the current structure of nanotechnology-related industry. As part of preparing this paper, thirty-four companies were identified for which information was available and which had nanotechnology related activities (see Appendix A for the list and note on sources). Five companies have strategies closely related to development of MNT assemblers. Thirteen other companies mainly provide equipment, consulting or contract research and development for nanoscale analysis and fabrication. Sixteen companies produce nanotubes, powders, ceramics and other nanoscale materials. Most of the companies provide materials, equipment or contract R&D, all of which could potentially be compatible with either closed or open source intellectual property. Only a few companies (notably CALMEC) have strategies explicitly focused on creating intellectual property and then obtaining revenue from licensing, though many others mention possible licensing or joint ventures. Many of the companies have benefited from government support, particularly through the U.S. National Science Foundation's Small Business Innovation Research (SBIR) program. This suggests that if U.S. government policies continue to favor proprietary intellectual propertym rather than public domain or open source, then these policies might significantly influence industry structure. None of the identified companies yet has a business model explicitly incorporating open source strategies.

Research business. While commercial development will become increasingly important over time, current activities are better characterized as nanoscience. Most of the relevant research and development is being done by university-based groups, much of it supported by public funding. Scientific values of sharing and publication encourage making new knowledge widely available. Pragmatic considerations may guide researchers to use open or closed source software depending on which seems most useful. However, aspirations of researchers and their universities to profit from intellectual property rights encourage closed approaches. Open source Unix programs have their roots in the Berkeley Unix operating system software, publicly funded and issued under an open source license at a time when US government policies emphasized making the results of publicly-funded research freely available to the public. Policies regarding intellectual property rights from government-funded research now strongly favor commercialization of intellectual property. Government laboratories are also under pressure to work with private corporations and use intellectual property rights as part of commercializing discoveries.

Industrial research. Notable exceptions to the primary role of universities in nanoscience are IBM, especially its Zurich lab, and Lucent's Bell Labs. Both organizations are prolific patenters, and are taking leading roles in some aspects of nanoscale science. However both currently seem to fund nanoscience as part of basic research, rather than out of immediate commercial goals. They both also have business models which are not primarily reliant on rents from monopoly intellectual property rights. Instead they emphasize providing integrated systems of hardware, software and services, with core competences that center on large scale implementation of advanced technologies.

Access for R&D. Open source licensing strategies might contribute to research and development which better facilitates scientific progress, and encourages commercial development to be done in ways which contribute to continued scientific advance. At present there is some tendency and tolerance for researchers to use patented techniques without permission, as part of research. However, as commercial development comes closer this risks problems if rights to the necessary patents cannot be obtained. If a critical mass of open source intellectual property relevant to molecular nanotechnology were developed, equivalent to the free versions of Unix, then this would offer security that applications could be smoothly implemented, rather than being dependent on the preferences and business tactics of a monopoly rights holder. In cases where universities and businesses conclude that their immediate and continuing needs for access to intellectual property outweigh speculative gains from royalities on intellectual property, then open source strategies would be the logical choice. To put this in less abstract terms, such willingness to choose open source is what happens already in decisions to buy Linux workstations or to use free software for molecular modeling.

Business opportunities. To the extent that any conclusions can be drawn at this early stage, it appears that current industry structure shows the potential for open source approaches, even though few companies appear to show an strong focus on open source approaches so far.[9] Consortia of industries interested in ensuring access to nanotechnology might find open sourcing a desirable strategy, especially if they represent a broad spectrum of interests or want to work in ways that do not elicit antitrust concerns.

Software

Molecular modeling is a key tool for nanotechnology. Drexler (1986) noted the potential for substantial design work to be done even before assemblers become available. Some work has been done along these lines, much of it presented in earlier Molecular Nanotechnology Conferences. The architectures within which various programs are combined could have a significant impact on the potential for open sourcing nanotechnology.

Programs. A large number of software programs are relevant to nanotechnology, ranging from detailed quantum mechanics calculations to computer-aided design (CAD) software. Many such programs have been placed in the public domain or released as open source. Given the dominance of academic researchers at the current stage of nanoscience this is unsurprising, but does show the foundations for further open source efforts. The Antas website lists over 200 software items "useful to the computational chemist" divided into 18 categories (Manunza 2000). The Scientific Applications for Linux website lists 357 items under "Chemistry, Biology and Related" software (Kachina Technologies 2000). For 67 items (19%), the list identifies whether the software is commercial, shareware or GPL. Of those so identified, 30 (45%) are commercial software, while 36 (54%) are listed as GPL, constituting 10% of the items in the list. Interest is growing in open source approaches to science and scientific software (BNL 1999, Bioinformatics.org 2000, Wilson 2000).

MNT Tools. A number of molecular modeling tools have been used by groups specifically involved in efforts to develop MNT. CavityStuffer, a pre-alpha experimental tool for packing predefined spaces with branched polymers, was developed at the Institute for Molecular Manufacturing and released under the GPL (Krummenacker 1996). Zyvex, "the first MNT startup," released DiamondCAD as "an experiment in open software development" (Zyvex 1999). The package does not seem to include any licensing statement, and the web page says Zyvex is "giving away" the program, which seems to mean that it has been put into the public domain. Version 0.3 links to two commercial software packages, SolidWorks and HyperChem. An earlier version of DiamondCAD, version 0.2, which is independent of these is also available. The web page, last updated January 1999, invited modifications to be submitted, but reveals no responses. Zyvex takes a pragmatic approach to mixing open source tools (e.g. Python) with available commercial software, as indicated by its website, its actions and statements from its CEO.[10]

The NanoCAD design system is designed for mechanical modeling of molecules, and intended to promote development of molecular nanotechnology (Ware 2000). NanoCAD was originally released under the GPL, with a later Java applet released under a "Berkeley-esque" license, i.e., similar to BSD. The NanoCAD mailing list had 1134 messages between October 18, 1995 to August 15, 2000 from about 45 participants, indicating a significant but not overwhelming level of activity.

Among the efforts discussed on the NanoCAD list was OpenChem (2000), conceived as an open source CAD system for "investigating nanotechnology, molecular structures, machines, and phenomena." OpenChem was intended to be Open Source, but licensing does not seem to have been clearly defined. The OpenChem project currently does not seem to be active. OpenChem drew on code from NanoCAD and other packages, including RasMol.The DisMol package was also developed from NanoCAD and RasMol (McCluskey 1998). It is labeled as "freeware" and the ReadMe file includes no licensing information.

Rasmol is a public domain package for molecular visualization developed by Roger Sayle (Martz 2000). RasMol's e-mail list had 480 members as of September 1999. Protein Explorer, released in summer 2000 incorporates code from Rasmol and from Chime, a proprietary package from MDL Information Systems, whose object code may be freely used by individuals, but which is not open source and not freely redistributable. In September 2000 an OpenRasMol website was established, with information on the release of version 2.6 under a license which allows modifications, but requires them to be disclosed and explained (Bernstein 2000). However OpenRasMol uses an idiosyncratic license with provisions which could give rise to confusion and cause incompatibility with other licenses.[11] The case of RasMol, past and present, seems to show how lack of clear and consistent licensing could lead to software development "forking" into different, inconsistent directions. The OpenRasMol initiative is intended to promote synthesis among various different programs which have been developed based on RasMol (Bernstein 1999). The history of RasMol seems to illustrate problems that can arise from simply putting a program into the public domain, and why release under a well-designed open source license could be preferable.

Early versions of the Fungimol (Freeman 2000) extensible system for atomic-scale design have recently been released under the LGPL. The system is intended to develop into a tool for MNT design work. The program architecture is designed to enable modification and extension using plug-ins. As discussed above, the LGPL permits combining code under the LGPL with other modules, so allowing development of proprietary plug-ins. The Fungimol package includes a detailed licensing notice about the terms of use for code incorporated in the program and reveals careful attention to requirements for proper licensing.

Systems architecture. Ideas about MNT systems architecture seem to be in an early stage of development. At the molecular level there are still disagreements about how and when it is appropriate to abstract to simpler approaches than detailed quantum calculations. At higher levels, massive parallelism and complexity pose major challenges beyond current software systems architectures. The distribution of computing capacity among different levels is an issue (which could be linked to to Merkle's (1994) proposal for a broadcast architecture). There is recognition that levels and modularity will be necessary, but so far few proposals analogous to such (once upon a time controversial) software abstractions such as high level programming languages, the Unix kernel or the internet transport layers. Almost all experience with computers suggests that such abstractions, even ones which initially appear to have a high cost in terms of narrowly defined efficiency, can make major contributions to the productivity of software development. Even such exceptions as the stack-based Forth language, and inclusion of assembly-language within C programs, "prove the rule" by their relatively minor role and marginalization to specialized uses. The extent to which useful abstractions for modules and levels are proposed and receive some consensus support in systems architectures for MNT could have a major impact on the opportunities for open source development. This need not take the form of a "cathedral-type" masterplan for systems architecture, but instead might develop more productively through simple protocols sufficient to link different components in "evolvable systems" (Shirky 1998).

Opportunities for open source. If analogies from existing open source apply, then the best prospects for open source may lie in several areas. Infrastructure-type services, e.g. applying widely known algorithms and other knowledge to common activities, may gain greatly from agreement on shared standards implemented through open approaches. To the extent that safety is an issue, then it may be important to assure the public and regulators that safety requirements have been complied with, and disclosing source is a very transparent way to do so. Customized design of nanotechnology products could occur, even if some of the underlying technology is closed. In part this would be analogous to the way desktop publishing and other tools have greatly expanded the potential for people to directly produce what they want, rather than having to work through specialized intermediaries. A mix of open and closed software could be encouraged through software architectures which allow plug-ins, macros, APIs and other enhancements. The most likely outcome is not one of everything being open source, but one of coexistence, cooperation and competition between open and closed source.

Homebrew hardware. While access to computers is widespread, equipment for working on the nanoscale is not as easily available. There was an initiative to develop a Homebrew STM (Rice 1995), intended to cost less than $1,000, which now seems defunct. The Simple STM Project (Alexander 2000) aims to build a simple scanning tunneling microscope that can image individual atoms for under $100. This is being developed by someone with substantial experience with scanning tunneling microscopes. The Homebrew STM relies on some patented designs, and so is not itself open source, but is an example of how at least some kinds of hobbyist access might evolve. Another ongoing STM Project (Muller 2000) has a materials budget of about $1100. The impact of "homebrew" development in leading to personal computers in the 1970s obviously gives grounds for speculation about, or at least not dismissing, major consequences from hobbyist development if it proves feasible. The ability of programmers to afford powerful computers may be one of the factors which has helped energize open source software development during the 1990s (Bennett 2000), and access to software and hardware could similarly shape the future growth of open source MNT.

AFMs and workstations. At present, atomic force microscopes and other professional nanotechnology tools are still expensive equipment sold individually to scientific laboratories and industry. The cost of hardware for most MNT-related tools is still well out of the hobbyist range. However, hobbyist type access to tools may be available to graduate students and those who have access to atomic force microscopes and other equipment as part of their studies and work. Potentially, this could offer parallels to the roots of GNU Unix, Emacs and other "Free Software" in university and corporate mainframe computers and workstations. Dependence on relatively expensive equipment is not a prohibitive barrier to the development of an open source community, though expanded access would create more favorable conditions.

Designs for the future. Several pioneering efforts have been made to apply open source approaches to software tools for nanotechnology, and a few efforts launched for hardware affordable by hobbyists and others without funds for expensive equipment. None of the MNT-oriented efforts has yet "taken off" into development of a mature and widely used application, or a large open source community.[12] Release of some packages into the public domain reveals a willingness to share results without limitations. No clear consensus seems to exist regarding licensing as a means to encourage others to contribute to the effort. However the use of the LGPL for Fungimol offers a useful precedent, as does the attention to creating an architecture suitable for use with plug-ins. Continued progress in other areas of nanotechnology is likely to further stimulate open source efforts, as research programs grow and the potential benefits of such software become apparent to a larger pool of potential contributors. MNT development can also be expected to benefit from open source activities in more general purpose molecular modeling software. The ways in which architecture evolves could have major implications for whether MNT systems architectures tend toward being monolithic and proprietary, or towards open systems with modularity and levels encouraging productive collaboration and competition among combinations of open source and closed source components.

Safety

One of the somewhat counterintuitive arguments for open source is that it is safer than closed source. Reliability of complex systems, security against computer viruses and other attacks, and integrity of cryptographic secrecy in communications all benefit greatly from peer review and other key elements of open source development. These advantages may also apply to nanotechnology. Talking about open sourcing nanotechnology may evoke fears about giving easier access in the future to those who might abuse the technology. Both these issues make it important to discuss the relationship between open source and safety.

Risks of not developing nanotechnology. For the long term, safety is a crucial issue for the development of molecular nanotechnology. However much of the discussion has concentrated on a few speculative scenarios of mature nanotechnology with hostile release of destructive replicators (gray goo), rather than a more comprehensive approach to the range of safety issues affecting an emerging technology.[13] Aside from potential nanotech "weapons of mass destruction," most safety issues will be analogous to more familiar and manageable risks such as fire, arson, car crashes, airplane accidents, electric shock and food poisoning. The major difference is that good design could eliminate or drastically reduce most of the risks for nanotechnology.[14] Good design might make nanotechnology inherently far *safer* than current technologies, thus potentially creating huge opportunity costs in terms of safety and other values if nanotechnology were *not* deployed.

Premises. To keep within the scope and length limits of this paper, discussion of safety in this section will be framed in terms of working hypotheses which seem generally accepted in most current thinking about risks from nanotechnology:

1. Good design could make nanotechnology safe in ordinary uses.

2. Nanotechnology could not be kept secret for long.

3. Defenses will be needed against use of nanoweapons.

Designing for safety. Despite initial worries, it has become increasingly clear that safety could be designed into ordinary applications of nanotechnology (Drexler 1990). Most products made with nanotechnology would be incapable of self-replication. Those able to replicate would usually depend on specialized inputs of materials, information and energy, without which they could not replicate. The Foresight (2000) Guidelines on Molecular Nanotechnology recommend additional measures such as designing in dependence on artificial vitamins and specialized energy sources, and requiring encrypted authorization in order to operate. In a Broadcast Architecture (Merkle 1994) individual nanomanipulators would depend on instructions from an assembler control computer and so would not be able to operate independently.

Safety through open sourcing. As discussed earlier, open source offers potential advantages for reliability through peer review to identify and correct problems.[15] This increases the chances that software and hardware will behave as desired, and so can be designed to be safe. Design efforts can try to ensure that if problems do occur then consequences are minimized, making software and hardware fail-safe or fail soft. Stability in the interaction of complex interrelated systems is one of the main advantages of open source versions of Unix. It is likely that open source nanotech could gain the same advantages in terms of reliable performance. Under open source it would be possible to check that safety precautions had been properly incorporated into design. Open sourcing would build on the efforts of a large community to identify problems and solve them. Given the impossibility of predicting all sources of possible failure, an open source community would be better able to rapidly respond to problems and learn how to prevent them. Thus, open source could potentially make valuable contributions to increased safety in ordinary applications of nanotechnology.

Dangers of increased availability. Analysts of nanotechnology's dangers have assumed that it would not be feasible to keep nanotechnology secret (Drexler 1986, Gubrud 1997, Freitas 2000). In contrast to initial speculations about technology that might possibly mature within a very short period of time, Kaehler's (1996) analysis convincingly argues that normal engineering learning processes are likely to constrain technical development. Technical innovation is likely to stretch out over a period of decades or longer, at least as long as advanced artificial intelligence is not available. While nanotech pioneers might be able to keep ahead of others, the scientific knowledge would be widely available, and any secrets would be vulnerable to spying and reverse engineering. The conclusion is that nanotechnology would become accessible to nations, groups and individuals who might use nanotech weapons for killing and destruction. Open sourcing would not create a new kind of threat. However it might make information available more quickly and so might increase the pool of individuals and groups able to use the technology. The most evocative, though far-fetched, vision of such risks is the specter of a lone sociopath able to release nanomachines which could reduce the entire biosphere to gray goo.

Designing to prevent misuse. Unless proactive measures are taken to promote safety, open source approaches might make it easier to bypass, remove or otherwise undo safety features designed into nanotechnology. However, a strategy for designing safety can protect against abuse, particularly from individuals and groups without advanced technical skills across the range of technologies required for nanotechnology (again, at least as long as advanced artificial intelligence is not available). Simple measures such as requiring authentication and monitoring from a remote source might be sufficient to block unsophisticated attempts to misuse nanotech. If "artificial fuel sources and artificial vitamins" are essential for the workings of the available models for assemblers, than redesign to use other mechanisms, or to produce them locally would require substantial additional expertise and effort. Changing from a broadcast architecture relying on an external computer to designs with onboard processing would require even more challenging design and engineering. If open source licenses require compliance with safety guidelines, and software packages and servers hosting them only include such software, then the best and most available tools will have safety built in, in ways that would require substantial time, effort and skills to overcome. The Foresight (2000) Guidelines recommend that:

> *Governments, companies, and individuals who refuse or fail to follow responsible principles and guidelines for development and dissemination of MNT should, if possible, be placed at a competitive disadvantage with respect to access to MNT intellectual property, technology, and markets.*

Licensing requirements could be one tool for enforcing such restrictions on access to MNT intellectual property. This is one case where the principle of forcing derivative works to fully comply with licensing conditions of the original work would be strongly justified. Such measures cannot eliminate all risks of abuse, but could make it substantially more difficult and unlikely. The rise of the internet and its associated open standards are an outstanding example of what can be achieved through open processes for self-governance, relying on initiative, sharing of information, rough consensus and standards-making through "requests for comment." Similar approaches could help the community of those concerned with MNT R&D to better address safety and other issues.

Enhancing active shields. The prevailing, though still speculative, conclusion has been that since nanotech weapons would ultimately be developed, an immune system-type technology of "active shields" would be needed for defense. Open source would offer the same sort of advantages for active shields against nanoweapons that it does for other design activities, by enlarging the pool of ideas and experience used to improve defenses. In the cases of computer viruses and cryptography, there are strong arguments that the advantages of sharing information about viruses, encryptation algorithms and other security arrangements more than offset the increased risk engendered by wider knowledge of vulnerabilities and tools for making attacks. The situation for active shields might be analogous, although it is far too early to know for sure. To the extent that this was uncertain or that closed source offered potential advantages, a pragmatic approach would be to employ a mixed strategy using both open and closed source systems in parallel, especially given that defenses would almost certainly use strategies emphasizing diverse and redundant systems.

Accelerating other safety strategies. Several other strategies have been suggested for dealing with the threat of nanoweapons. Advanced artificial intelligences might be developed and tasked to ensure safety (Yudkowsky 2000). Ubiquitous placement of tiny video cameras might enable monitoring as part of a "transparent society" (Brin 1998). "Truth Machines," (Halperin 1997) working like advanced lie detectors, might enable detection and control of dangerous individuals, including those in leadership positions. Open source could potentially accelerate the development of hardware and software needed to implement such strategies, if controversies about their technical feasibility and ethical desirability were resolved by agreement to pursue such strategies.

Open and closed levels. Analogous to current computer systems, open source applications might direct an assembler, whose internal software and hardware might be closed source. The assembler level kernel could include hardware and software for ensuring compliance with safety codes, including checking on the authenticity and integrity of higher level programs before they are allowed to operate. Higher level programs could include built-in safety characteristics, just as the Java programming language has built-in restrictions on the operations which it can perform. Designed-in restrictions might include requirements for authentication and authorization before operation, as well as reporting on location, operations and outputs and other monitoring information in accordance with safety protocols.

Nanoblocks and other limited nanotech. Even more limited forms of nanotech have been proposed and could enable safe access. Limited assemblers would be capable of making only a restricted range of products, and incapable of self-replication (Drexler 1986). In contrast to the smart, general purpose "utility fog" proposed by Hall (1996), it might be possible to make "nanoblocks" with limited capabilities, adequate to be formed into a variety of useful structures by specialized nanofabricators, guided by a programming language with built in restrictions, analogous to the Java "sandbox."[16] Such nanoblocks and other nanotech products might be made in specialized secure facilities, and then distributed for subsequent use. It is worth noting that even if access to assemblers were restricted, nanoblocks and limited assemblers could facilitate the creation of an open source community safely innovating designs for nanotechnology.

Advantages for safety. Safety concerns about molecular nanotechnology need to be addressed. In the short to medium term, open sourcing offers significant advantages for designing safety into ordinary applications of nanotechnology. In the longer term, nanotechnology cannot be kept secret, so dangers are largely independent of whether the technology is open or closed, while open sourcing could contribute to defensive techniques and other means for increasing safety.

Conclusions

Open MNT initiatives. One of the distinguishing features of open source software development has been the demonstration of how much can be accomplished by the initiative of interested communities, without waiting for additional government support or policy reforms. The initiatives of diverse individuals and groups have played central roles in open source software development, and development of open source molecular nanotechnology could well be similarly diverse.

Licensing. Several strategies might be useful in promoting open MNT. Agreement on core principles, similar to those in the Open Source Definition, could reduce conflicts and encourage commercial participation. Given the importance of physical structures in nanotech, open sourcing of nanotechnology needs to address patent rights in more detail than do current licenses, perhaps drawing on precedents from current initiatives seeking to develop open computer hardware and open publications. In some cases, patent pools with an incentive arrangement of restricted rights might help balance commercial and community interests. Most of the goals of open source could be obtained with simple licenses such as the X11 and BSD licenses. For those who want to ensure disclosure of modifications, while still allowing modular linkage with closed software in combined works, the LGPL may be a suitable license.

Business models. Companies, entrepreneurs and investors would benefit from recognizing the ways in which open source methods can synergize with business models relying on providing goods, equipment, and expert services such as customization and specialized research and development. Open source business models are likely to be more successful in a dynamic environment, less risky and more robust against unanticipated changes, compared to strategies highly dependent on closed intellectual property. One promising opportunity for exploration would lie in forming consortia for open MNT R&D, supported by companies which want to ensure access to nanotech as a production technology. Open source methods could help avoid antitrust problems for such consortia.

Molecular engineering software. Software for MNT design seems the most promising area for further open MNT efforts. Systems architecture for software, and eventually nanodevice hardware, which incorporated suitable abstractions such as modularity and levels would help promote fruitful competition and combination of open and closed source software and hardware.

Safety. The proposed Foresight Guidelines on Research in MNT could create an important opportunity for strengthening self-governance within the MNT R&D community, at a minimum by stimulating discussion of when and how safety should be addressed. Open sourcing offers important advantages for promoting safety in MNT design.

Topics for further study. A number of areas could benefit from further research on the applicability of open source principles for MNT.

- Study of a larger sample of nanotechnology-related companies and software packages would permit more thorough analysis of current industry structure and licensing strategies.

- Further analysis of patent licensing and pooling arrangements could help clarify the current situation and options for the future.
- The relevance of open source as an alternative to private monopoly could be analyzed in terms of the strength of increasing returns, network externalities and other factors favoring the evolution of software monopolies, and how these would apply in the case of MNT.
- Lessons about safety from computer security, cryptography and industrial design could be explored in greater depth, analyzing the impact of different intellectual property arrangements.
- Impacts and policy options concerning intellectual property generated in government-subsidized research deserve further analysis.
- The relative advantages and disadvantages of open versus closed intellectual property for different goods and services could be further assessed for the case of nanotechnology.
- Building on insights about the value of open source as a producer good, discussion and scholarship on open source could pay more attention to the dynamics of industrial consortia for research and development, especially those which seek to create open standards.
- Comparative analysis of coordination in open source programming projects with the somewhat more formally structured institutions for internet self-governance might yield ideas useful for implementing safety guidelines, and for coordination of broader efforts to open source MNT.

Notes

Author information. Bryan Bruns holds a Ph.D. in Development Sociology from Cornell University. He has an independent practice as a consulting sociologist, based in Chiang Mai, Thailand. He is a member of the International Association for Public Participation and the International Association for the Study of Common Property, a Fellow of the Society for Applied Anthropology and a Senior Associate of the Foresight Institute. Views expressed in the paper are those of the author and do not represent any organizations with which he is or has been affiliated.

Responsibility for any technical bloopers and other errors and omissions lies with the author.

Acknowledgments. In addition to the sources specifically cited, this paper has benefited from discussions of open source at the Foresight Senior Associate Gathering, May 19-21, 2000 and from online discussions on www.nanodot.org and the NanoCAD mailing list. In a posting on the usenet sci.nanotech newsgroup on Jan 5 1995, Sean Jackson (sdj1@cityscape.co.uk) suggested that nanotech designers could share their ideas via the same kind of public source licenses used for software, mentioning Linux, and Emacs and posting a brief version of the Free Software Foundation's license. This appears to be the first published suggestion that Free Software licensing be applied to nanotech. Christine Peterson, the Executive Director of the Foresight Institute participated in the discussions which resulted in the earlier Debian Guidelines being developed into the Open Source Definition, and is credited with coining the term "Open Source" for this software.

In comments on an earlier paper, Robin Green pointed out the importance of distinguishing between public domain status and open source licensing. Eric Raymond provided helpful comments in correspondence and conversation. Correspondence with Tim Freeman, author of the Fungimol software package, helped focus my attention on the relevance of the LGPL as a suitable license for MNT. Will Ware, author of NanoCAD and moderator of the NanoCAD mailing list, commented on the educational role of open source and on other issues. Eugene Leitl responded to several questions posted with an earlier draft. He and several others on the NanoCAD list contributed to interesting threads on systems architecture, vat manufacture and collaboration. Robert Nansel's comment on Nanodot pointed out the URL for John Alexander's Simple STM Page. John Alexander clarified that his STM uses patented technology, and gave a pointer to Jurgen Muller's STM Project. Chris Phoenix and Pat Gratton made helpful comments on drafts. A draft of this article was discussed on Slashdot on October 19, 2000 under the title "Open Source Nanotechnology." The Slashdot article was based on an editorial in Nanotechnology Magazine that cited the draft paper and recommended use of open source methods in the development of molecular nanotechnology.

Updates

Draft 0.4 posted on September 1, 2000 and announced on Nanodot and on NanoCAD list.

Draft 0.5 September 8. Reorganized with summary at beginning. Added discussion of antipatents, systems architecture and homebrew hardware.

Draft 0.6 October 16. Changed titles. Miscellaneous minor revisions, including following up on comments received.

Draft. 0.7 Reformatted references. Revisions regarding what not to open source, value of simple X-type licenses, potential of open source in infrastructure, safety and customization, and need for research on factors influencing software monopolies. Deleted (incomplete and somewhat inconsistent) investment and staff columns in nanocompanies list.

Draft 0.8 Converted to Microsoft Word. Reformatted endnotes and section headings.

[1] This paper largely follows the concepts, assumptions and scenarios discussed by Eric Drexler's (1986) in *Engines of Creation.: The Coming Era of Nanotechnology*. The major differences are assumptions that:

a. Safety can be designed into ordinary applications of MNT (per Drexler 1990).

b. Learning processes characteristic of other engineering R&D will constrain the rate of technological advance in MNT (Kaehler 1996).

c. MNT will continue to advance more rapidly than either general purpose artificial intelligence or radically self-improving computer design programs. This assumption is more debatable, but postulating strong artificial intelligence tends to shroud discussions of the future behind the veil of an unknowable technological "singularity" (Vinge 1993), while this paper focuses on intellectual property policies and strategies for MNT research and development in the short to medium term.

[2] Eric Raymond's essay originally appeared online in 1997, and was subsequently revised, and followed by further online essays on "Homesteading the Noosphere" and the "The Magic Cauldron." These were published in book form in 1999 and references in this paper are to the 1999 print version.

[3] In a series of articles in the Foresight Update Newsletter, Elizabeth Enayati gave a useful introduction to intellectual property issues affecting molecular nanotechnology.

[4] Explicitly putting code in the public domain would fulfill the minimum requirements for the open source definition of making the source code available and allowing modification and free redistribution. (Eric Raymond in a personal communication agreed that this satisfies the open source definition "in a trivial sense.")

[5] Descriptions of licenses and links to the webpages for the licenses can be found on the Open Source website (www.opensource.org) and the Free Software Foundation website (http://www.gnu.org). The online version of this paper also has direct links to sites with further information on the licenses.

[6] The most recent releases of the Python software are under a new license, which is intended to be compatible with the GPL, but has not yet been accepted by the Free Software Foundation. Apparently this is because the Python license specifies the venue in which disputes would be considered, which is not specified in the GPL.

[7] Information on the Open Hardware initiative is at www.openhardware.org. For examples of licenses see the Open Collector (opencollector.org/hardlicense/licenses.html) and Open PPC Project (www.openppc.org/licenses.html) websites. Another interesting hardware initiative is the GNU book project, which has a special GNU++ license (www.gnubook.org).

[8] Defensive publication by IBM was explained by P Wilson in a March 14, 2000 note on "Publishing ideas" on the oreillynet.patents newsgroup.

[9] Under current intellectual property conditions, a mixed strategy able to work under both open and closed IP might be a pragmatic choice. Celera's strategy in biotechnology includes both speculative patent claims and a core business model based on providing specialized information services (as well as backing from a business whose revenue streams come from selling equipment).

[10] Zyvex's website discusses open source and makes clear that it uses both open and closed source tools including Python. Posts by Jim van Ehr, Zyvex's CEO, on the NanoCAD mailing mailing list clearly state a pragmatic attitude towards the use of open source.

[11] The license for OpenRasMol is said to be "GPL-like," so that any code derived from the OpenRasMol code would also be under the same license. The licensing notices with the package is not clear about the creation of combined works, but in correspondence H J Bernstein has said that these would be permissible (though discouraged, as with

the LGPL). Use with nuclear facilities or in connection with aircraft is prohibited, so the license would not comply with the Open Source Definition principle on no restrictions on "fields of endeavor." The language of the licensing notice appears to restrict it to "personal use," i.e. may prohibit commercial use or commercial distribution of the software, though the authors state their intention to give permission for those who request to redistribute the package on CDs or websites. This idiosyncratic license shows many examples of the kind of diverse, confusing and potentially incompatible licensing requirements which could be prevented by using the GPL, LGPL, MPL, X11 or other standard Free Software or Open Source license.

[12] As part of trying to promote more "design ahead" to be able to take advantage of molecular assemblers once they arrive, Robert Bradbury has proposed a Nano@home project, including options to encourage resulting designs be open source (www.aeiveos.com/~bradbury/Proposals/NanoAtHome.html) . A distributed application for Folding@Home was mentioned on Nanodot (nanodot.org/article.pl?sid=00/09/29/2031204) and discussed on Slashdot (slashdot.org/science/00/09/26/2111216.shtml)

[13] For a recent discussion of gray goo and other threats from hostile replicators, see Freitas 2000.

[14] The ways in which shielding and interlocks make microwave ovens far safer than gas ovens offers an example of the potential for good design to build in safety and convenience.

[15] Security and open source is discussed by Eric Raymond (1999:155). The topic of open source and security frequently comes up periodically on the Slashdot forum. A recent example of discussion about risks from easier availability of tools which might be used in attacks can be seen in commentaries on "Security Through Obscurity a GOOD Thing?" July 7, 2000. slashdot.org/articles/00/07/27/1343236.shtml Advantages and disadvantages of code having code exposed for inspection were discussed in commentaries on "Security Focus Responds to ESR [Eric S. Raymond] Column on OSS [Open Source Software] Security, April 17, 2000 http://slashdot.org/articles/00/04/17/0817211.shtml

[16] Nanoblocks were suggested by Eliezer Yudkowsky in personal communications with the author.

References

Alexander J D 2000 *Simple STM Project* www.geocities.com/spm_stm

Bennett J C 2000 *The Anglosphere Century: The Future of the English-Speaking Nations in the Internet Era* (Rowman & Littlefield) at Press

Bernstein H J 2000 *Home Page for OpenRasMol Molecular Graphics Visualisation Tool* www.OpenRasMol.org

Bernstein H J 1999 RasMol -- *Managing the Divergence of Variants in an Open Source Multiple Platform Environment* openscience.bnl.gov/Abstracts/RASMOL.shtml

Bioinformatics.org 2000 *The Open Lab* bioinformatics.org

BNL-Brookhaven National Laboratories 1999 Open Source/Open Science Conference openscience.bnl.gov

Brin D 1998 *The Transparent Society : Will Technology Force Us to Choose Between Privacy and Freedom?* (Reading, MA: Addison-Wesley)

Brooks F P 1995 *The Mythical Man-Month: Essays on Software Engineering. Anniversary Edition.* (Reading, MA: Addison-Wesley)

Bruns B R 2000 *Nanotechnology and the Commons: Implications of Open Source Abundance in Millennial Quasi-Commons* presented at the Eighth Conference of the International Association for the Study of Common Property, Bloomington, Indiana, USA, May 31-June 4, 2000

Coase R H 1990 *The Firm, the Market and the Law* (Chicago: University of Chicago Press)

Drexler K E 1986 1990 *Engines of Creation*. (New York: Anchor Books) http://www.foresight.org/EOC/index.html

Drexler K E 1990 *Afterword* in *Engines of Creation*. (New York: Anchor Books) http://www.foresight.org/EOC/EOC_Afterword.html

Drexler K E 1999 Senior Associate Letter Letter #28 (Palo Alto, CA: Foresight Institute)

Doll J J 1998 The Patenting of DNA *Science* **280** 689-690 http://www.sciencemag.org/cgi/content/full/280/5364/689

Ellenbogen J C 1997 *Matter as Software* presented at *Software Engineering and Economics Conference* April 2-3, 1997 (McLean, Virginia: Mitre Corporation) www.mitre.org/technology/nanotech/downloads/Matter_As_S-W.pdf

Freitas R A 1999 *Nanomedicine Vol. I. Basic Capabilities.* (Austin, TX: Landes Bioscience)

Foresight Institute 2000 Guidelines on Molecular Nanotechnology www.foresight.org/guidelines/current.html

Freeman T 2000 *Fungimol* www.infoscreen.com/fungimol

Freitas R A 2000 *Some Limits to Global Ecophagy by Biovorous Nanoreplicators with Public Policy Recommendations* (Richardson, TX: Zyvex) www.foresight.org/NanoRev/Ecophagy.html

Gubrud M A 1997 *Nanotechnology and International Security* presented at the Seventh Conference on Molecular Nanotechnology.

Hall J S 1996 *Utility Fog: The Stuff Dreams are Made of* in *Nanotechnology: Molecular Speculations on Global Abundance* B C Crandall (ed) (Cambridge, MA: MIT Press) pp 161-184

Halperin J 1997 *The First Immortal* (New York: Ballantine)

Hardin G 1968 The Tragedy of the Commons *Science* **162** 1243-1248. http://www.sciencemag.org/cgi/content/full/162/3859/1243

Hardin G Extensions of "The Tragedy of the Commons" *Science* **280** 682-683. http://www.sciencemag.org/cgi/content/full/280/5364/682

Hargrave R and Malamud C 2000 *Transparent Patents* voice.media.org/essays/patent.html

Hecker F 2000 *Setting Up Shop: The Business of Open-Source Software* Draft revision 0.8 http://www.hecker.org/writings/setting-up-shop.html

Heller M A and Eisenberg R S 1998 Can Patents Deter Innovation? The Anticommons in Biomedical Research Science **280** 698 - 701 http://www.sciencemag.org/feature/data/980465.shl

Kachina Technologies 2000 *Scientific Applications for Linux: Chemistry, Biology and Related* sal.kachinatech.com/Z/2/index.shtml

Kaehler T 1996 *In-Vivo Nanoscope and the "Two-week Revolution* in *Nanotechnology: Molecular Speculations on Global Abundance* B C Crandall (ed) (Cambridge, MA: MIT Press) pp 49-60

Krummenacker M 2000 *Problems with the Current Intellectual Pseudo "Property" (IPP) System, especially Patents* http://www.n-a-n-o.com/ipr/fi-gathering2/ipp-scenarios.html

Krummenacker M 1996 *CavityStuffer* www.ai.sri.com/~kr/nano/cavstuf/cavstuf.html

Manunza B 2000 *Antas: Software list and links* antas.agraria.uniss.it/software.html

Martz E 2000 *Molecular Visualization Freeware: Protein Explorer, Chime & RasMol* klaatu.oit.umass.edu:80/microbio/rasmol

McCluskey P 1998 *Dismol: A freeware molecule display applet including source code* www.rahul.net/pcm/dismol/DisMol.html

Merkle R C 1994 *Self replicating systems and low cost manufacturing* in *The Ultimate Limits of Fabrication and Measurement* Welland M E J.K. Gimzewski J K (eds) (Dordrecht: Kluwer) pp 25-32 www.zyvex.com/nanotech/selfRepNATO.html

Muller J 2000 *STM Project* www.e-basteln.de/index.htm

Olson M 1971 *The Logic of Collective Action: Public Goods and the Theory of Groups* (Cambridge, MA: Harvard University Press)

OpenChem 2000 *OpenChem* www.openchem.org

Open Source Initiative 2000 *Open Source Definition* http://www.opensource.org/

Ostrom E 1990 *Governing the Commons: The Evolution of Institutions for Collective Action* (Cambridge: Cambridge University Press)

Perens B 1999 *The Open Source Definition* in *Open Sources: Voices from the Open Source Revolution* (Sebastopol, CA: O'Reilly)

Peterson C 1999 *Inside Foresight* in *Foresight Update* **38** 2

Raymond E S 1999 *The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary* (Sebastopol, CA O-Reilly)

Rice J 1995 *NanoTools: The Homebrew STM Page* atom.snu.ac.kr/stmwebpage.html

Shewmaker M 2000 *The Open Patent License* www.openpatents.org

Shirky C 1998 *In Praise of Evolvable Systems* www.shirky.com/writings/evolve.html

Software Patent Institute 2000 *Database of Software Technologies* www.spi.org

Vinge V 1993 *The Coming Technological Singularity: How to Survive in the Post-Human Era* www-rohan.sdsu.edu/faculty/vinge/misc/singularity.html

Ware W 2000 *NanoCAD in Java: A freeware nanotech design system including source code* world.std.com/~wware/ncad.html

Williamson O 1996 *The Mechanisms of Governance* (New York: Oxford University Press)

Wilson G 2000 *Alchemy or Science: The Open Source to a Scientific Enlightenment* in *Scientific Computing World* http://www.scientific-computing.com/cgi-bin/editorial_frameset.cgi?pg=/cgi-bin/featurepage.cgi%3ffeat=37%26pg=1

Yudkowsky E 2000 *The Plan to Singularity* http://singinst.org/singularitarian/PtS/plan.html

Zyvex 1999 *Zyvex DiamondCAD* www2.zyvex.com/DCAD/Software.html

Appendix 1. A brief list of some "Nano" companies and their products

See notes below for explanations

	PRODUCTS	(planned and current)			
COMPANY	**Equipment/Hardware**	**Software&IP**	**Services**	**Notes**	**Founded**
A. MNT-ORIENTED					
Atomasoft		NanoSim/ Nanosoft OS	design, simulation, engineering and consulting services	MEMS and photonics	1997
Molecular Manufacturing Enterprises			seed capital, advice, contacts, and other support services	links to IMM	1998?
Molecular Robotics	taggants, nanotubes, MEMS, AFM, quantum computers			enabling technologies for MNT, (Institute of Nanophysics)	1999?
Technanogy			incubator, invests money, expertise and intellectual property	May invest in aligned ventures. Said to be space-oriented.	2000?
Zyvex	Zybot manipulator,Rotapod assembler, MNT assembler			links to U. Texas (Dallas) and others	1997
B. R&D AND EQUIPMENT					
CALMEC California Molecular Electronics Corporation		patents and trade secrets: technology licenses: fees and royalties	contract R&D	strategy to pursue IP	1997
Hewlett-Packard	computers and equipment				1939
IBM	computers and other IT	software	consulting	leader in research	1911
Kaweenaw Nanoscience Center			consulting, contract R&D	quantum optics & nanotech, laser	2000

				scissors and tweezers	
Mitre			research for government clients	molecular electronics. Nanosystems Group since 1992	1958
Lucent/Bell Labs	communications systems			does relevant research, no specific nanotech program	Bell Labs 1925
MEC - Molecular Electronics	DRAM and other molecular electronics equipment?			cooperation with Rice, Yale & Penn State Universities	1999
Moore Nanotechnology Systems	ultra-precision machine systems			some DARPA-developed technologies	≤1998
Nanogen	microchip for biological analysis: cartridges and workstations	licensing and joint ventures	contract research	child of Nanotronics	1993
Nanolab	devices based on carbon nanotubes	pending patent	collaborative research		2000
Nanovation	photonic components		optical circuit design (Apollo Photonics)	formerly US Integrated Optics	1996
NanoWave		licensing	contract R&D	position encoders	1995
SDL Queensgate Nanopositioning	subsystem OEM			nano-positioning and sensors	1979
C. MATERIALS					
1. Argonide	powders		contract R&D	applying Russian technology	1994
2. CarboLex	single-walled carbon nanotubes				1998
3. DEAL International Inc.	carbon nanotubes		R&D	no website	
4. eSpin	polymer fibers and webs		technology development		
5. Hyperion Catalysis International	Graphite Fibril(TM)				

	nanotubes				
6. Invest Technologies	metal nanopowders			Russian technologies	1997
7. Materials Modification Incorporated			contract R&D	nanolayer coating process	1986
8. Nanocor	nanosized clay minerals for plastic resins				1995
9. Nanomat	nanocrystalline materials and nanostructures		consulting and technical assistance	Pennsylvania and India	≤1999
10. Nanomaterials Research Corporation	powders and derivative materials	possible patent licensing			1994
11. NanoPierce	nanoparticle electrical connections			website down	
12. NanoPhase Technologies Corporation	powders and engineered products			moving toward large [tons] production capability	1989
13. NanoPowders Industries	precious and base metal powders				1994
14. NextTech Materials	nanoscale ceramics for fuel cells	licensing and joint ventures			1995
15. Physical Sciences Inc.	developing carbon nanotube membrane for fuel cells	licensing	contract R&D	technology development company	1973
16. Powdermet	powders and other materials		contract R&D		1996

Table notes

Sources: Compiled August 5-8, 2000, from websites listed on the Nanotechnology Industries list of companies (www.homestead.com/nanotechind/companies.html), or under industry on the Loyola College Nanotechnology Database (itri.loyola.edu/nanobase/). Companies with inactive websites (e.g. Nanotechnology Development Corporation) or inaccessible websites (e.g. Nanoprobes) not included, nor are companies whose websites indicate little explicit focus on nanotech (e.g. NanoLogic) or that seem no longer active in nanotech (e.g. Xerox). These two lists were chosen as well-informed sources that provided a manageable sample which could be studied with limited research resources. Two nanotechnology companies were added later, Molecular Electronics, and Technanogy.

The lists used do not include most suppliers of molecular modeling software (discussed separately in the paper), STM/AFM equipment and other relevant goods and services. For a list of equipment suppliers, see the EMBL website (www.embl-heidelberg.de/~altmann/companies.html#INSTRUMENTATION). The Nanotechnology Industries list of tools includes software and equipment companies not in the list above (www.homestead.com/nanotechind/tools.html). Nanotechnology potentially affects almost every company, especially those producing any kind of material goods. For longer lists of nanotechnology-related companies, (which could be studied for a more comprehensive assessment) see the Atomasoft website (www.atomasoft.com), and the Nanotechno posting on RagingBull (www.ragingbull.altavista.com/mboard/boards.cgi?board=NANOTECH&read=274). Robert Freitas' Nanomedicine page lists nanomedicine research organizations and companies (www.foresight.org/Nanomedicine/#NMResComOrg).

Products: Includes both current and projected (possibly vaporware) products. Software/IP listed only if for external sale/licensing, not just on company's own equipment/hardware products.

Comments and corrections are invited to BryanBruns@BryanBruns.com