調査資料

米国の政府支援研究開発プロジェクトにおける知的財産権の取り扱いの変遷の歴史とその背景

平成１１年３月

財団法人　日本情報処理開発協会
先 端 情 報 技 術 研 究 所

本調査資料は、平成10年度にMUSE Associates（MUSE社）に調査委託し、
入手した基礎資料やデータを取りまとめたものである。

Introduction

米国の政府支援研究開発プロジクトにおける知的財産権の取り扱いの変遷の歴史とその背景

米国政府が支援するソフトウェアの研究開発において知的財産権の取り扱い方によって米国産業の振興に大きく寄与してきたのでははないかという仮説のもとに、わが国の産業振興策を進める上で知的財産権の取り扱い方に大きな示唆を与える以下の８つの問題意識に整理した。本調査は、この８つの質問に回答するものである。

1．米国は、政府支援研究開発プロジェクトの成果に対し、原則公開の開放的政策を取ってきた。この政策が、民間でのソフトウエア開発の基盤生成に大きく寄与したのではないか。
2．米国では、政府支援研究開発プロジェクトの成果を原則公開、企業の商業化を通じ税金回収という形で投資を回収するという姿を原点としつつ、徐々に発明者、著作者に与える権利を拡大してきた。このような動きの中で、政府、民間Ｃｏｎｔｒａｃｔｏｒｓ、個人の発明者・著作者の利害はどう調整され、そこに流れる基本思想は何か。
3．既存のアイデア、ソフトウェアを含み込む形で、ソフトウェアとは開発されるものである。そのような場合、著作権の設定はどうなっているのか。政府支援プロジェクトではそうした事態にどう対処しているか。またプロジェクトの成果であるソフトウェアの商業化のルールはあるのか。
4．ソフトウェア産業のビジネスモデルは、自社開発ソフトの販売というもっとも単純なもの、他者に先駆けて自社のソフトウェアを公開してデファクトスタンダードを確立し、その後から付加価値の高いビジネスを展開するもの、開発当初からソースコードを公開してしまうオープンソース等、ますます複雑になってきている。こうした変化は政府支援プロジェクトにおける知的財産権戦略に何か影響を及ぼしているのだろうか。
5．複数企業が協同で行う政府支援プロジェクトの場合、知的財産権の取り扱いは企業間でどのようになっているか。
6．上記の問題と関連して、成果物が特にソフトウェアの場合どうか。
7．米国政府は政府支援研究開発プロジェクトの成果物として、ソースコードの納入を要求するのか。
8．上記５，６，７を総合して考え、知的財産権のルールがうまく規定されていない場合、せっかくの政府支援プロジェクトの成果であるソフトウェアが死蔵され、商業化に生かされないことになる。これでは税金の無駄使いとなってしまう。米国ではどのように対処しているのか。

目　次

第1章　政府支援により開発されたソフトウエアの知的財産に関する米国のシステム
The US system of federally-funded software IP

A　知的財産保護のシステム……6
Systems of intellectual property protection

B　知的財産とソフトウェア……9
Intellectual property and software

C　政府支援により開発されたソフトウェアの知的財産……20
Federally-funded software IP

D　政府支援により開発されたソフトウェアの歴史的な成功例……33
Federally-funded software historical successes

E　オープンソースのうねり……41
The Open Source movement

第2章　米国政府支援により開発されたソフトウェアの
知的財産の扱いに関する8つの質問に対する回答……45
Eight key questions

第3章　日本に示唆するもの……57
Conclusions and implications

付属資料……63
Appendix

第1章　米国の政府支援研究開発プロジェクトにおける ソフトウエアの知的財産権

The US system of federally-funded software IP

1	**The US system of federally-funded software IP**	
A	**Systems of intellectual property protection**	**p2**
B	Intellectual property and software	p9
C	Federally-funded software IP	p20
D	Federally-funded software historical successes	p33
E	The Open Source movement	p41
2	Eight key questions	p46
3	Conclusions and implications	p58
A	*Appendices*	p64
A1	*List of Sources*	
A2	*Glossary*	

Intellectual property (IP) law provides the framework within which the legal rights in the 'products of the mind' are created, allocated and enforced

- Intellectual property can be defined most generally as the intellectual products of the human mind
- A system of intellectual property law provides property rights (ie rules of ownership and permissible use) in these products
- The fundamental purpose of US IP law is to provide incentives to develop further innovations, or new IP
- The mechanism by which these incentives work generally follows the mechanism employed in the US system as a whole - namely economic incentivization
- Works not protected or not protectable under an IP system are said to be in the 'public domain' - in other words no one holds any rights in the IP, and no one can in any way restrict any member of the public from using that IP in any way they see fit

Elements of an IP system

1 **Definition of the subject matter to which the IP law applies**

2 **A set of requisites for protection, eg**

- What qualities the subject matter must possess to be protectable
- Who is entitled to assert the IP right
- What procedural steps must be taken to acquire or retain the IP rights

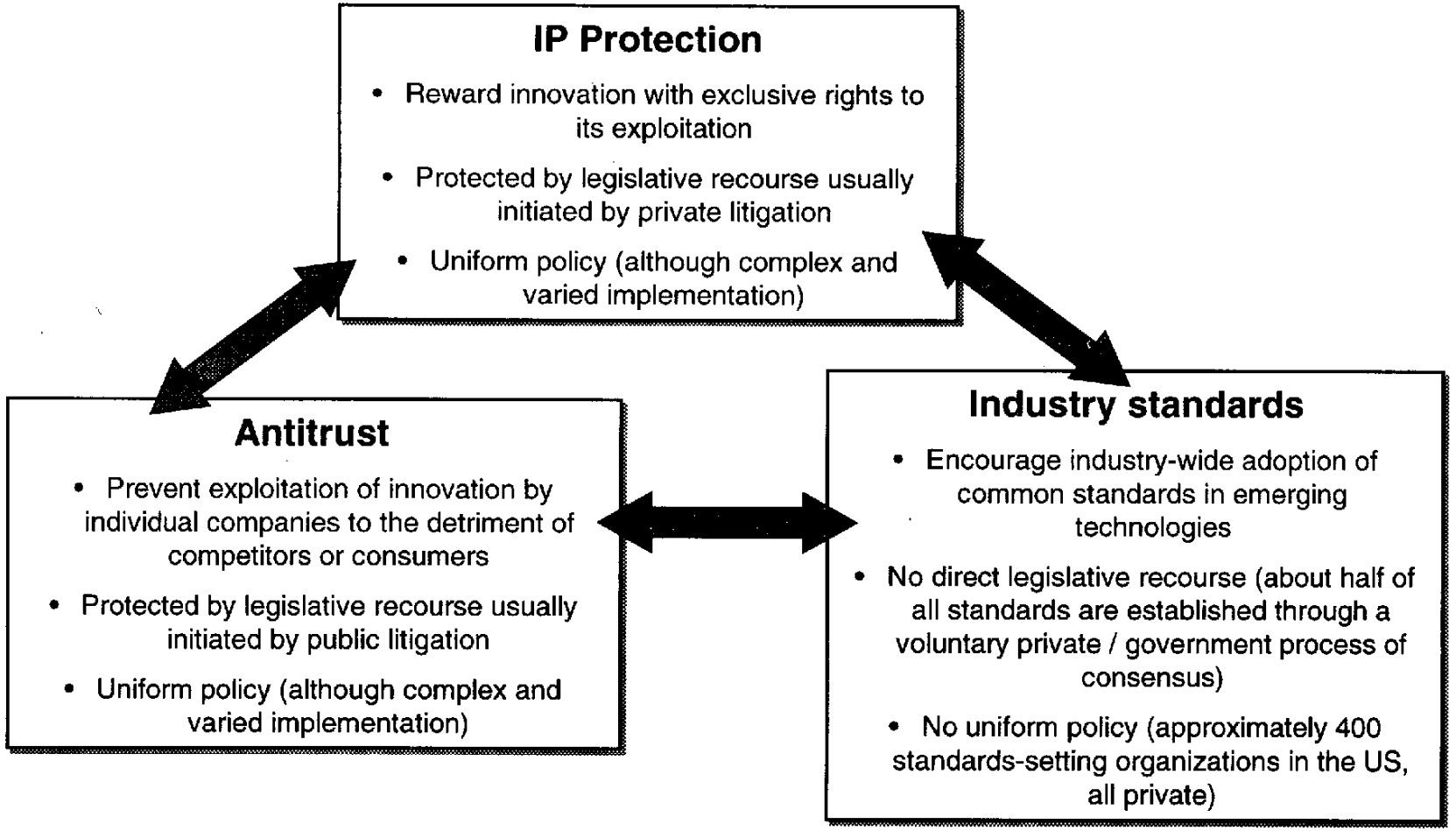
3 **A set of (exclusive) rights to exclude other people from certain activities**

4 **A public policy limitation on the extent of the owner's intellectual property rights**

5 **A procedure for determining whether infringement has occurred**

6 **A specification of the remedies available**

IP protection rests in a sometimes uneasy balance with industry standards and antitrust regulation



1	**The US system of federally-funded software IP**	
A	Systems of intellectual property protection	p6
B	**Intellectual property and software**	**p9**
C	Federally-funded software IP	p20
D	Federally-funded software historical successes	p33
E	The Open Source movement	p41
2	Eight key questions	p46
3	Conclusions and implications	p58
A	*Appendices*	p64
A1	*List of Sources*	
A2	*Glossary*	

Definition of Software for IP purposes

In practical terms, 'software' consists of source code, object code and documentation

The three components of software

- **Source code**
  - The program, generally written in a high level computing language
  - Underlying approach and actual algorithms are clear to the user
  - Technical documentation can be included directly in the source code
- **Object code**
  - The program, generally in an executable (machine language) form
  - Limited information about approach and algorithms can be determined
- **Documentation**
  - The supporting instructions for use, explanations , and notes, which gives away more or less of the program's secrets

- In the commercial arena, the three components of software are treated equally in the terms of a licensee's usage rights. Software is generally defined to include all three, and licensee's rights refer to 'software' rather than individual components
- In general, however, commercial entities selling pre-packaged software do not provide copies of source code (with the exception of so-called 'Open Source' software). Contractors developing work for hire do provide source code - this requirement is usually explicitly addressed by the parties' contract
- In practice, a commercial entity's view is typically that the best IP protection is non-disclosure of source code. Commercial licenses are consequently a 'right to use'; users are prevented from modifying software by the simple expedient of not knowing enough about it

Legal protection for software generally takes the form of 'copyright', 'patent' or 'trade secret'

- As a 'literary work' or 'writing', software is protected by copyright law. The Copyright Office began accepting source code listings as copyrightable subject matter in 1964; in 1980, Congress explicitly added 'machine-readable computer programs' to the Copyright Act
- As a 'machine' which 'processes', software which is 'new, non-obvious and useful' can be protected by patent. Software was generally believed not to be patentable until a landmark Supreme Court judgement in 1981, since which the number of software patents has begun increasing steadily
- As a 'compilation of information' which is 'used in one's business' and 'provides one with an opportunity for a competitive advantage' and also 'which is maintained as a secret', software itself, and / or documentation, and / or component tools, algorithms, techniques etc, can be held as trade secrets. Trade secrets are protected by 'common law' (ie case-by-case application of general laws, rather than the more specific 'statutory law') at a state level.

"Programs are not only text, they also behave; software is a machine whose medium of construction happens to be text. Creating programs is as a result simultaneously a work of authorship and a work of invention. In crossing that allegedly unbridgeable barrier software creates significant conceptual difficulties, conflicting as it does with assumptions that have long been part of the legal system"

Intellectual Property and Software
Randall Davis, MIT 1991

Several unique characteristics of software pose fundamental challenges to the IP protection system

Unique IP aspects of software

- The cost of software is almost entirely in development, with little or none in production / replication / distribution, and the barriers to IP abuse that exist in other industries such as automobiles or aircraft do not exist
- Software combines aspects of a 'literary work', where the actual form of expression is important, and a 'machine', where the process employed is important - this means that software is 'eligible' for both copyright and patent protection
- However, neither copyright nor patent protection for software keeps all stakeholders happy - a substitute generated by a competitor can be sufficiently different not to infringe copyright, and yet perform the same essential function; a patent defined too broadly can impinge on parts of software performing quite a different function as a whole; a patent defined too narrowly affords no protection at all
- Software is generally combined from discrete elements put together in new ways - hence IP protection can be misused to claim that other uses of the elements are infringements

Today copyright is the principal mechanism for protection of software IP

Qualification: for protection, software must be:

- Original (usually established on a case-by-case basis)
- Fixed in a tangible means of expression (eg written on paper, printed out)
- Non-utilitarian (ie 'not useful' - generally held to mean do more than convey information or display an appearance. However, software is generally considered an exception to this rule)

Authorship

- Only the author may claim a copyright unless the work is 'made for hire' - made by an employee within the scope of employment, or 'specially commissioned' - in which case the employer owns copyright
- Ownership can be sold, licensed, or given away

Procedure

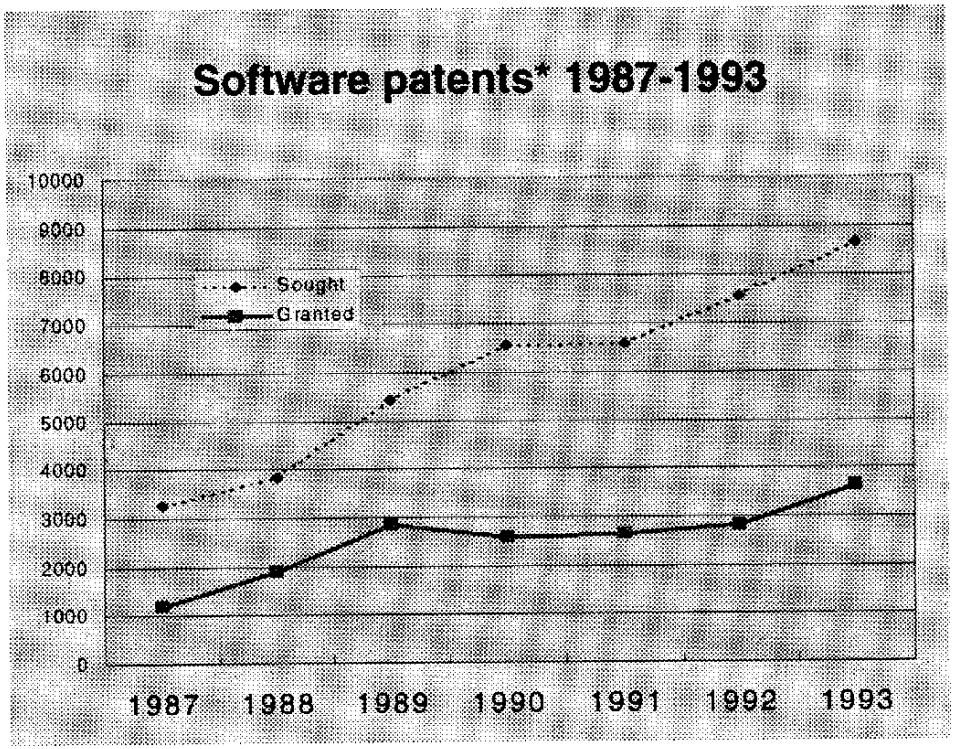
- Until March 1989, a copyright notice was required on the work; this is no longer the case
- Registration of copyright is not required, except to bring an infringement lawsuit
- Protection lasts the life of the author plus 50 years

Rights: copyright provides the author with five exclusive rights:

1 To reproduce the work
2 To prepare 'derivative works'
3 To distribute copies to the public
4 To perform the work publicly
5 To display the work publicly

- The three key exceptions are fair use (use of the work can sometimes infringe the above rights if the purpose is criticism, comment, reporting, teaching, scholarship, or research), first sale (an owner of a copy can resell or dispose of that copy), and private use (the owner of a copy can use it as he sees fit in the privacy of his own home)

Software patent protection is gradually increasing



* Total patents in classes 364 and 395, to which software patents belong. Note that some non-software patents are also included in these figures, although the percentage is not clear

Source: Software Patent Institute

- 11500 software patents were issued in 1997, and 80000 software patents could be in force by the year 2000 ('Owning the Future', Seth Shulman)

- Software can be eligible for either 'utility patents' (processes, improvements, ideas - but not algorithms alone, which are held to be 'expressions of laws of nature or science' - 17 year protection) or 'design patents' (appearance, 14 year protection)
- In return for public disclosure of his invention, the owner of a patent can exclude others from making, using or selling it in the US (or grant a license to do in return for royalty payments)
- In order to receive a patent, a software invention must be 'novel', have 'utility', be 'non-obvious', and be properly disclosed
- Most debate over software patents centers on the criterion of 'non-obviousness', which is assessed by the Patent Office which performs a search for 'prior art' (ie inventions and ideas in existence at the time of the patent which would have meant that the invention was in fact obvious to those aware of them)

However, software protection by patents is highly controversial and widely held to be detrimental to the software industry

- One of the biggest problems with software patents is that the US Patent Office does not have access to sufficient information to be able to determine accurately the state of the 'prior art'
- Most software is not patented, but held as trade secrets - ie it is not disclosed, and hence cannot be reviewed by the Patent Office; in addition, unlike the chemical, mechanical and biotechnology fields, the Patent Office does not yet have a centralized computer database of software patents, although they are being developed {eg by the Software Patent Institute and by IBM (the 'Intellectual Property Network')}
- The software industry is moving so fast, and attracting so much talent from so many different areas, that establishing whether or not an invention would have been obvious to anyone is an almost impossible task
- Software is so complex that establishing its purpose, and perhaps its hidden purpose, can be very difficult for patent examiners - Richard Stallman, the Open Source champion, testified at the 1994 Patent Office Silicon Valley hearing that a colleague of his had won a patent on an 1845 scientific theory by embedding it in a software program
- Those best qualified to establish prior art are often those accused of infringement - while this is effective in repealing inaccurate patents, it is costly for the companies affected

The cost of software patents to the industry

- In 1993 Compton's New Media was granted 41 patents which enabled them to claim royalties on virtually all multimedia CD-ROMs. After lengthy and expensive investigation, the patents were overturned
- Accepting that sooner or later they will face patent infringement claims, many companies 'fight back' by filing for patents they feel they could 'trade' (in cross-licensing agreements) if they were sued
- Many companies regularly spend millions defending patent claims
  - Adobe spent $4.5m and 3500 man hours of its principal scientist defending a 1992 claim
  - Autodesk spent $1m defending 17 claims over a period of five years
  - American Multi-Systems went out of business due to loss of customer trust after a patent suit (which they ultimately defeated)
- In 1997, IBM used its arsenal of software patents to secure cross-licensing agreements with 52 companies

Industry support for both copyright and patent protection for software is weak, except in certain key areas

IP Protection Methods: Industry Survey

Type of software function	Copyright	Patent	Both	Neither	Respondents
Source code	**88%**	2%	3%	8%	318
Object code	**65%**	2%	3%	27%	293
Pseudocode	37%	1%	1%	61%	278
Module design	18%	9%	1%	72%	269
Algorithms	9%	12%	1%	79%	303
User interface commands	6%	1%	0%	92%	294
Icons	43%	1%	1%	56%	307
User interface layout	19%	1%	1%	79%	302
User interface sequences	9%	9%	0%	90%	295
Look and feel	5%	0%	0%	94%	312
User interface functionality	5%	4%	0%	91%	300
Computer images	**81%**	1%	0%	18%	316
(Weighted) Averages	**33%**	**4%**	**1%**	**63%**	

Source: Siggraph, from 'The Impact of Intellectual Property Rights on US Competitive Infrastructure', NCMS 1994

In practice, protection for software is as much a function of 'non-disclosure' as it is of legal protection

The dual mechanisms of software IP protection ...

Market preservation

- By restricting the rights of users to further distribute software, the software author ensures that would-be users will have to obtain the software from him

Limit competition

- By minimizing competitors' access to the proprietary aspects of software, the author increases the time delay in which he can take advantage of his work

- A literary work, such as a novel, cannot readily be reproduced by another who simply reads it, since the production of a work based on the same idea requires literary skill equivalent to that of the original author
- However seeing software source code, and understanding its operation, is often enough for a skilled programmer to produce a copy that is sufficiently different not to risk copyright infringement litigation, but essentially 'copies' the ideas of the original
- Hence non-disclosure of source code is a critical element of software IP protection

A key issue for government contractors is therefore whether or not they are forced to disclose source code to the government, and whether or not the government will disclose it to others

Voices calling for radical reform in the US system of software IP protection are growing louder

- Industry players are increasingly complaining that the growth of software patents is wasting valuable resources that could be devoted to innovation
- Theorists are denouncing the current system of IP protection as inadequate for software, which has characteristics of IP traditionally protected by both patent and copyright, but which also comes dangerously close to 'fundamental concepts' which (according to IP law) should be available to all
- The Open Source movement is creating a new model of IP ownership and distribution, the eventual impact of which could be far-reaching (see Section E)
- Some view the openness of government IP as a threat to US international competitiveness, enabling foreign competitors to 'catch up for free'

"A new intellectual property rights framework is necessary, one that addresses the realities of competition in the next century"
The Impacts of Intellectual Property Rights on US Competitive Infrastructure *NCMS, 1994*

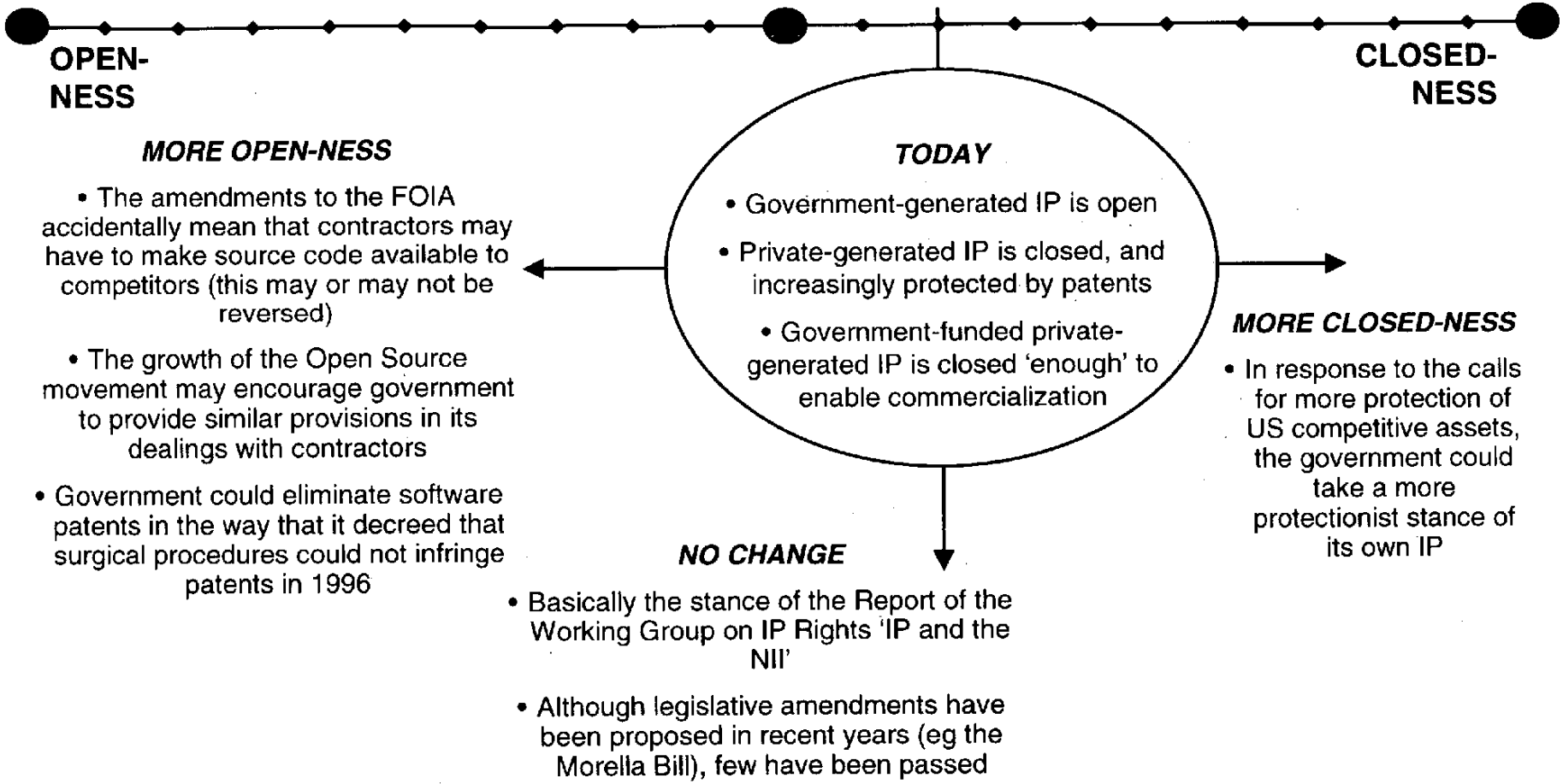
The US is fast approaching a "crisis in intellectual property law"
Pam Samuelson, *IP Law Professor, UC Berkeley, 1998*

"Patents as they stand now are a real problem"
Linus Torvalds, *inventor of Linux, 1999*

"The vast conceptual realm does not naturally lend itself to competitive market-based trade. Rather, ideas and concepts must be open to all."
Owning the Future, *Seth Shulman, 1999*

IP System Future Evolution

The US system of IP protection is almost certain to evolve in the near future, but the direction of evolution is unclear



1	**The US system of federally-funded software IP**	
A	Systems of intellectual property protection	p6
B	Intellectual property and software	p9
C	**Federally-funded software IP**	p20
D	Federally-funded software historical successes	p33
E	The Open Source movement	p41
2	Eight key questions	p46
3	Conclusions and implications	p58
A	*Appendices*	p64
A1	*List of Sources*	
A2	*Glossary*	

High-Level Government Policy

Official government policy at the highest level is that private contractors retain commercial rights in software developed under government contract

"Each agency head shall ... cooperate, under policy guidance provided by the Office of Federal Procurement Policy, with the heads of the other affected departments and agencies, in the development of a uniform policy permitting federal contractors to retain rights in software, engineering drawings and other technical data generated by federal grants and contracts, in exchange for royalty fee use by or on behalf of the government."

Executive Order No. 12591, 'Facilitating Access to Science and Technology'

signed by the President on April 10th, 1987

Government policy is not to directly promote the software industry, but rather to let the action of the free market exploit software IP

- The US government does not seek to promote the software US industry, but rather to enable the software industry to help itself
- In general, rights are available to contractors for software they develop, but they must proactively 'claim' them
- In complex situations, the contractors are able to negotiate up front - so that a fair rights allocation is agreed before work commences
- Wherever possible, the government is moving towards 'COTS' purchases, for which it receives only similar rights to any other buyer; as a monopsonist*, the government can provide a substantial market for software products in its own right - but it does not support failing companies; rather, it evaluates and purchases the best packages

"The US government does not have much of a policy of promoting the software industry, or intellectual property in general"

John Overton, past Chairman of ABA (American Bar Association) Committee 308 'Government Relations to Copyright'

* A monopsonist is a single, dominant buyer (a monopolist is a single dominant seller)

The government cannot hold copyright in works it creates, and all such works are in the 'public domain'+

- 'Putting the government in the software business' by reserving exploitation rights for the government is considered undesirable
- Since the federal government operates for the benefit of the people, as set out in the US Constitution, it cannot have a 'financial' interest in developmental technologies in the way that a business or financial institution can (and should) have (the only exception to this general principle would be when a national security interest requires protection against improper disclosure of information)
- The federal government cannot hold copyright in any works it develops itself, and all government works are deemed 'public domain'. Government can however be 'assigned' copyright by the actual author*
- Public domain works are made available to the public through clearing houses such as the General Services Administration's 'Federal Software Exchange Center'
- Government can ask a contractor to sign a 'work made for hire' agreement, in which case the government is deemed the author, and the work would be in the public domain. A 'work made for hire' agreement normally covers:
  - a work prepared by an employee within the scope of his or her employment
  - or a work specially ordered or commissioned (only under certain specific circumstances)

+ The case has been made that this weakens US international competitiveness, since the IP is freely available to non-US companies as well as to US companies (the Morella Bill of 1989 sought to allow government agencies to hold copyright, but was defeated in the House).

* An author can assign copyright to any third party by written agreement (for example a book author may assign copyright in a book to the publisher)

IP developed in the course of federal acquisition is governed primarily by the FAR and DFARS

- Intellectual property law provides a 'default setting' of rights allocation when software is created
- When IP is developed under a contract between parties, the contract can replace or supplement the rights policy of intellectual property law
- In the case of government acquisition, all contracts are subject to the policy of the Federal Acquisition Regulation (FAR), which has the force and effect of law
- Individual agencies have the authority to adopt supplementary regulations which can in some circumstances contradict or change the FAR policy
- As far as software IP is concerned, the most noteworthy supplement is the DFARS (Defense FAR Supplement), which is quite different from the FAR in its treatment of software IP

FAR

- Part 27, entitled 'Patents, Data and Copyright' deals specifically with the acquisition of rights in software and technical data
- Part 52 sets out the standard contract clauses (to be included in acquisition contracts) that implement this policy

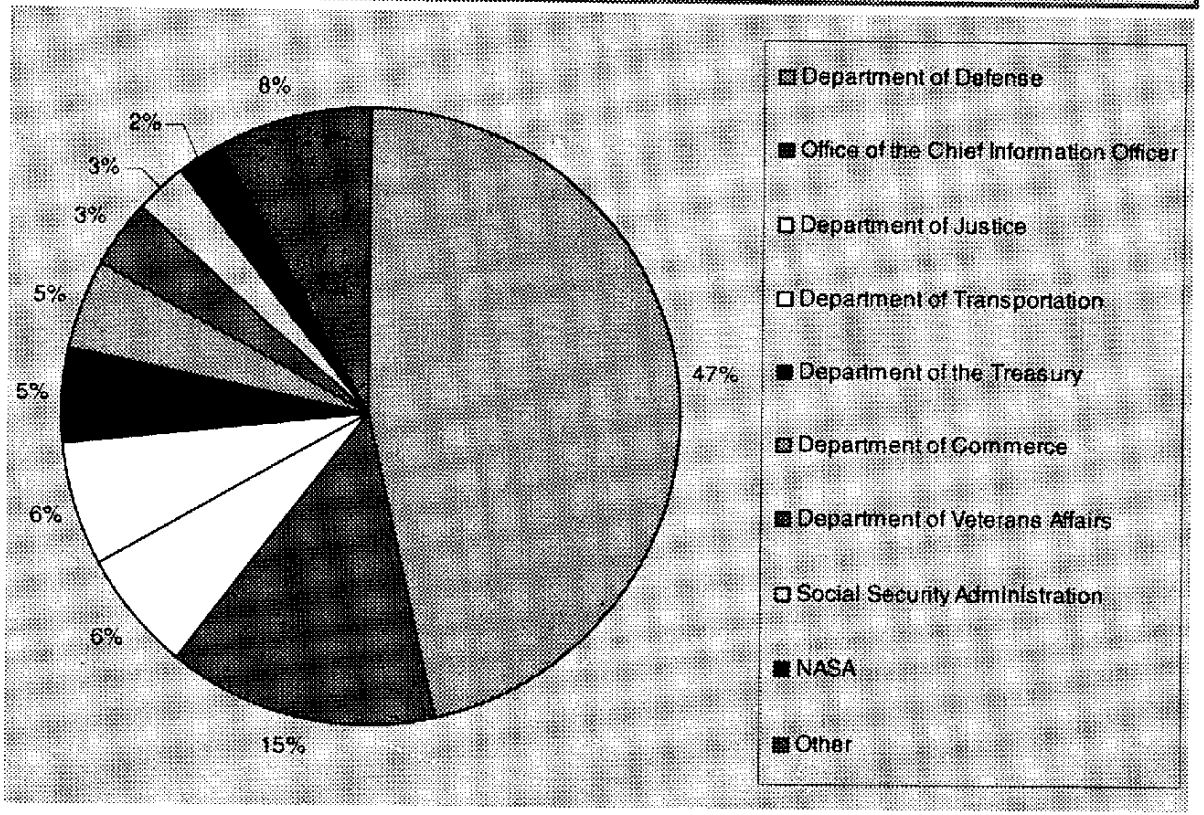
DFARS

- DFARS section 227 includes the section 'Rights in Technical Data and Computer Software' which substitutes for FAR 27 in DOD contracts
- DFARS section 252 contains the DOD standard contract clauses to be used in place of those included in the FAR

Agency Software Spend

The DOD accounts for about half of government software spend, with the various civilian agencies making up the remainder

Shares of 1998 federal ADP Equipment, Software, Supplies and Equipment spend of $5.5 billion



Source: Federal Procurement Data Center

- The FPDC (Federal Procurement Data Center) does not gather statistics on software funded as a by-product of contracts and grants ostensibly for different purposes
- Total government software spend is actually a much higher proportion of the $27 billion R&D spend, and of the some $200 billion total acquisition spend
- For example, studies of DOD spend alone found that software spend was around $12bn in 1985, and (expected to be) around $30bn in 1990*

* 'Comparative Studies in Software Acquisition', Steven Glaseman, 1988

The legal terminology used in FAR and DFARS leaves much room for debate over the treatment of source code

FAR 27.401 defines computer software to mean 'computer programs, computer databases and documentation'. Government rights to software apply equally to both programs and documentation. The term 'source code' is not used, but since source code must be either a 'computer program' or 'documentation', it is implicitly covered.

→ **FAR definitions do not explicitly distinguish source code, but the FAR treats rights to all elements of software in the same way**

In DFARS, computer software means 'computer programs and computer databases'. 'Computer programs' are defined in DFARS 227.421 as 'a series of instructions or statements in a form acceptable to a computer, designed to cause the computer to execute an operation or operations'.

Documentation is defined in DFARS 227.421 as 'technical data ...'. 'Technical data' is later defined as 'recorded information, regardless of the form or method of recording, of a scientific or technical nature'

→ **In DFARS definitions, source code could be treated as either a 'computer program' or as 'documentation' ('technical data'), and rights treatment - including the ability to negotiate rights treatment in individual cases - is consequently open to interpretation and variation on a case-by-case basis**

Government rights to software purchased or developed using government funds take one of three forms

Unlimited Rights	Restricted Rights	Government Purpose License Rights
• Use the software (FAR and DFARS) • Duplicate it (FAR and DFARS) • Release it (DFARS) • Disclose it (FAR and DFARS) • Prepare derivative works of it (FAR) • Publicly display and perform it (FAR) • Pass the above rights to third parties (FAR and DFARS)	• Use the software with the computer for which it was obtained (FAR and DFARS) • Use it with a backup computer if the first computer becomes inoperable (FAR and DFARS) • Copy it for backup purposes (FAR and DFARS) • Modify it and combine it with other software (FAR and DFARS) • Disclose it and reproduce it for use by support service contractors (FAR) • Use it with a replacement computer (FAR)	• Use the software • Duplicate it • Disclose it to support service contractors under an NDA*
Notes • Unlimited rights are not the same as ownership, because they are not exclusive • They are also not the same as placing the work in the public domain, since (whether FAR or DFARS) the government receives only a subset of all possible rights	Notes • The government is allowed to use the software for its original purpose, and to use it in further (originally unforeseen) government work	Notes • The government is allowed to use the software for the purpose for which it was originally intended only • Government Purpose License Rights are not explicitly so-named in the FAR, but are the approximate equivalent of the rights obtained if a contractor claims copyright • Under DFARS, Government Purpose License Rights are available for 'technical data', but not for 'computer software'

* NDA = Non-Disclosure Agreement

Government Rights Obtained

The type of rights the government obtains for software depends on the sources of funding, funding agency, and individual contract negotiations

	Civilian contract (covered by FAR)	Defense contract (covered by DFARS)
Software developed by government using government funds	• No copyright exists in work, and work is 'public domain' (no rights are attached to the work)	
Software developed by private entity using government funds	• By default, government obtains 'Unlimited Rights' but does not own copyright • Contractor has option to claim copyright in the work. If contractor claims copyright, and the government grants it, the government receives only 'Government Purpose License Rights' • Copyright is generally sought and granted • Government can request contractor to 'assign' copyright to government	• Contractor automatically receives copyright, in which case government generally receives 'Restricted Rights' in 'programs', and 'Government Purpose License Rights' in 'technical data' • Government can append the 'special works clause' (included in DFARS) to the contract, in which case the government vests copyright*, and holds 'Unlimited Rights'
Software developed using private funds ('COTS' = Commercial Off the Shelf)	• Government generally obtains 'Restricted Rights' • Government can negotiate for greater rights	• Government generally receives 'Restricted Rights' in 'programs', and 'Government Purpose License Rights' in 'technical data' • Government can negotiate greater rights • Government can receive less than the set of Restricted Rights only by a procedure of 'permission to deviate' (but procedure is cumbersome and rarely invoked)
Software developed using 'mixed funds' (some commercial parts, perhaps modified using government funds; some government-funded parts)	• Government generally obtains rights equivalent to 'Government Purpose License Rights' • The 50% point (based on % of total cost) is generally the threshold for mixed funding treatment (industry funding below 50% receives the same treatment as 0%)	• No separate treatment

* The government is prevented by § 105 of the Copyright Act from vesting a copyright in a work (ie 'claiming' a copyright in a work) - it is, however, allowed to own previously-existing copyrights assigned to it.

This means that a copyright supposedly vested by the government according to the 'special works' provision may not be enforceable in a Court of Law.

This legal uncertainty has not been resolved or tested in the Courts.

Software developed using mixed funding is generally the most controversial area, and much uncertainty over rights policy remains

- Mixed funding software, which relies on in some fashion software that has already been written (either commercially-available privately-funded software, or software developed under a prior government contract, or both) forms an increasingly large proportion of software developed under government contract
  - Software is getting more complicated
  - The introduction of the 'Modular Contracting' acquisition strategy in the Information Technology Management Reform Act of 1996 is a direct encouragement for contracting officers
- Contractors using software they intend to sell commercially must be very careful in their usage of the software, and in their contract negotiations, to preserve the restricted rights that would pertain to the commercial software in its unmodified form
  - The DFARS provides that commercial software procured from a third party is safe - 'those portions of the derivative software incorporating restricted rights software are subject to the same restricted rights' (227.471)
  - The DFARS states that 'only that portion of the resulting product in which the original product is recognizable' (227.481) will be eligible for restricted rights. This means that for example 'modular additions' to source code would leave the original code intact, and hence recognizable. Line changes, however, would be subject to (arbitrary) judgement.
  - Negotiation in defense contracts is only possible over technical data rights (which cover documentation, but not necessarily source code)
  - The FAR requires that when the commercial software costs less than 50% of the total, it will receive the same treatment as the software under development. This cost assessment is often very difficult to make, and consequently somewhat arbitrary. In practice, however, the commercial contractor generally retains copyright, in which case the government receives government purpose rights in the entirety of the software project
  - Negotiation is however possible in every instance - it is expressly permitted by FAR 27.404

Universities, historically the largest recipients of federal funds and generators of IP, hold a significant balance of power in IP rights negotiations with the government

- The concept of the 'free market' goes beyond the economic value of ideas: it defines the roles and relationships of all the stakeholders in the system of research and idea production

- For example, universities, who receive a large proportion of the total Federal funding for research and development, and generate a large proportion of the US's 'new' IP, are powerful and important stakeholders: university policies, as much as government policy determine the eventual fate of IP
  - The University of California*, in its 'Contract and Grant Manual' states that 'the publication policy requires that, with limited exceptions, the University will undertake research or studies only if the scientific results can be published or otherwise promptly disseminated ... the University, in general, must own the research results created under sponsored research agreements'

- Historically, the power and proactive use of IP by universities has meant that their policies have had as much of a role in shaping US policy (including government regulations) as government has itself

* The University of California comprises 9 campuses, 5 medical schools, 3 law schools, manages 3 GOCO national laboratories, and wields an $11bn budget

Government non-disclosure

Government is bound by the FOIA to disclose federally-funded information to the public, but often avoids the issue by not requiring contractors to disclose the information to it in the first place

- Under the Freedom of Information Act (FOIA), the Government is required to disclose certain information to any member of the public that requests it, barring certain exceptions in the interests of national security¤
- This mechanism can be used by companies to obtain copies of any data* (software) supplied to federal agencies under federal contract, as well as any data produced by federal employees (regardless of who owns the 'copyright')
- In the tradition of 'prevention is better than cure', therefore, one of the key mechanisms for protecting software and other data is simply for a contractor not to disclose the entirety of the data to the Government sponsoring agency
- A 1995 study of a number of DARPA project recipients found that DARPA's flexibility in not requiring exhaustive copies of data was key to their subsequent commercial use+

¤ the DOD regularly posts and then pulls data from its website - the most recent example was in September 1998, when the DOD removed detailed architectural blueprints of the Global Command and Control system

* software is considered data for the purposes of IP legislation, except when explicitly defined otherwise

\+ 'Participant views of Advanced Research Agency Other Transactions', 1995, Institute for Defense Analyses

The government, perhaps unintentionally, is currently embroiled in a controversy concerning new 1998 legislation to open federally-funded research data

- A rider in the Fiscal Year 1999 Omnibus Appropriations Act, passed in 1998, requires the Office of Management and Budget to modify Circular A-110* to state that 'data related to published research findings' will now be covered by the Freedom of Information Act (FOIA), and consequently must be made available to requests from members of government or the public
- The OMB published its proposed revisions on February 4th 1999, but is accepting public comments until April 5th; the issue has raised widespread concern, and has been taken up by numerous interest groups for the scientific community, in particular the AAAS+
- The revisions would affect non-profits (including most universities) engaged in grant-funded or cooperative research projects, but not private contractors
- The original intention of the rider was entirely well-meaning - to enable beneficial public access to government data which was originally intended for public benefit. For example, if the government conducts a survey of nursing homes or banking facilities, then (the theory goes) that raw data should be made directly available to the public to inform their choices
- As yet, the OMB has not explicitly defined 'data', and much concern centers both on what may be included in this definition (such as software source code), as well as the possible abuse of the provision by 'interested parties' to obtain valuable research data for free, and the consequent possible negative effect on government - industry partnerships

* Uniform Administrative Requirements for Grants and Agreements with Institutions of Higher Education, Hospitals, and Other Non-Profit Organizations. OMB Circulars are executive branch implementations of legislation intended to provide implementation guidelines for federal agencies usually for a period of two years

\+ American Association for the Advancement of Science, the world's largest interdisciplinary federation of scientists

1	**The US system of federally-funded software IP**	
A	Systems of intellectual property protection	p6
B	Intellectual property and software	p9
C	Federally-funded software IP	p20
D	**Federally-funded software historical successes**	p33
E	The Open Source movement	p41
2	Eight key questions	p46
3	Conclusions and implications	p58
A	*Appendices*	p64
A1	*List of Sources*	
A2	*Glossary*	

Government research has made a vital contribution in particular to the development of protocols and software sub-components, many of which play a role in industry end-user applications today

- Government research has been directly involved in few of the key software markets today (personal computer operating systems, personal computer productivity applications, databases, cryptography, enterprise resource applications)
- However the primary reason for this is that commercial products such as these play a relatively small role in research, and conversely the programs that are most useful in research are unlikely to find mass markets (such as weather simulators, astronomical orbit calculators, atomic particle simulators etc)
- The two key areas of overlap are:
  - algorithms, subroutines, sub-components, protocols, programming languages - ie the 'building blocks' of almost all software
  - certain specific applications which have managed to find wider applications (see chart on right)

Application area	Role of government IP
Data analysis	• Work on pattern recognition by eg CIA, NSA, FBI has led to a variety of data collection and analysis tools - databases, data warehousing, data mining, personalization
Design automation	• Many CAD / CAM / CAE tools emerged as a byproduct of aerospace development
Internet protocols and tools	• Some direct contributions (eg Telnet, Gopher) • Many indirect contributions (eg Mosaic, Yahoo, Inktomi)
Mobile / wireless	• Ad-hoc networking, spread spectrum, GPS
Operating systems	• Some Real Time Operating Systems (eg VxWorks)

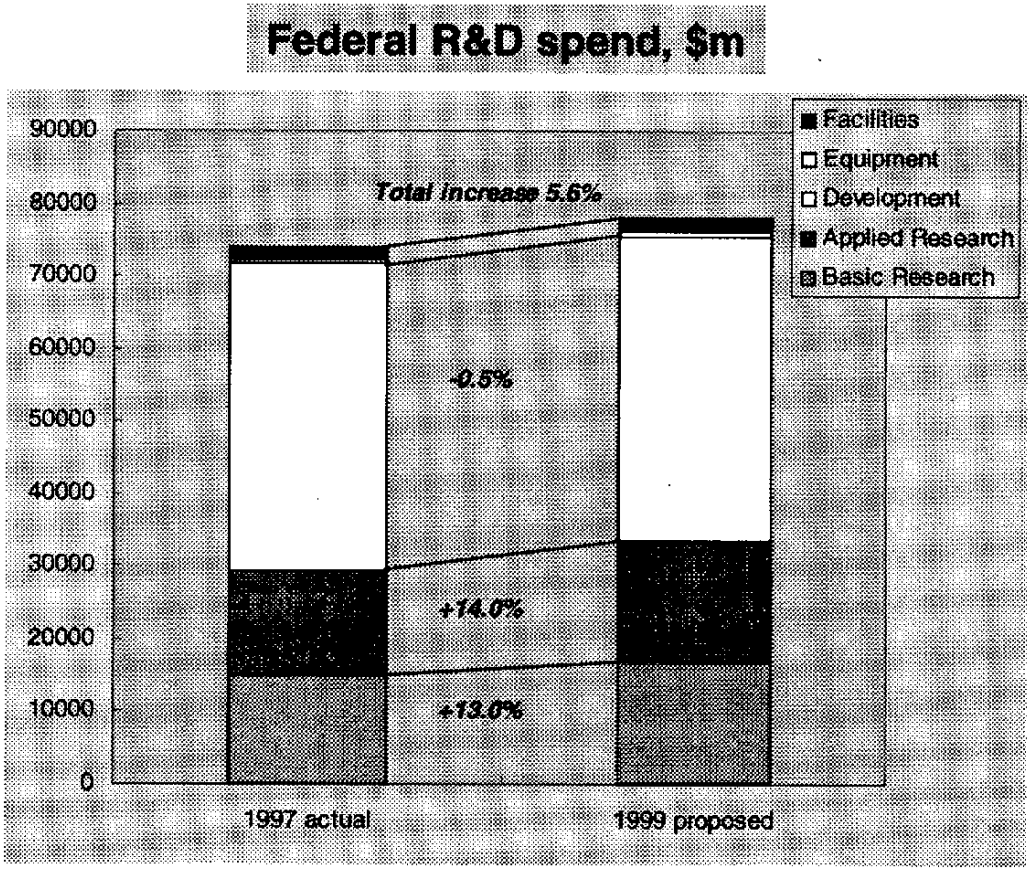
Government funding has played an even more significant role in the many software applications in use in the high-performance computing and academic world

- Many hundreds of application-specific software programs are in use in the academic community - a few key examples of which are shown in the chart on the right
- The majority of them were produced in the course of prior research, often as by-products to the specific aims of other research products
- The copyright in the software was generally held by the university at time of production (irrespective of the source of funds - most of which came from Government sources)
- The universities then often assigned the copyright to a commercial entity (often formed by the author) in return for rights to use and distribute the software internally

Astronomy, Astrophysics and Cosmology	AIPS++, AIPSview, COSMICS, CACTUS, KRONOS, MIRIAD, TITAN, ZEUS-2D, ZEUS-3D
Chemical Engineering	ASCEND IV, Aspen Plus, Aspen Dynamics/Custom Modeler, gPROMS, SpeedUp
Computational Chemistry	ADF, AMBER, CADPAC, CASTEP, CERIUS2, CHARMM (Harvard), CHARMm (MSI), CRYSTAL 95, DISCOVER, DMOL3, FAST STRUCTURE, GAMESS, GAUSSIAN94, GAUSSIAN 98, Insight II, Jaguar, MacroModel, MOLCAS, MOLPRO, NWCHEM, Q-Chem, UHBD
Computational Fluid Dynamics	FIDAP, CFX, FLUENT, FLUENT-UNS, Nekton, RAMPANT, STAR-CD, STAR-HPC
Database Visualization	AVS Express, SGI MineSet, Visible Decisions Discovery
Knowledge Discovery	Clementine, DB Intellect
Mathematics / Statistics	AXIOM, BLACS, BLAS, Complib, DAGH, FFT library, FFTW, IMSL Fortran library, LAPACK, MATLAB, Mathematica, NAG Mark 17, SMP, PARPACK, PETSc, PINEAPL, PLAPACK, SAS, ScaLAPACK, SCSL, S-PLUS, SPRNG
Structural / Solid Dynamics	ABAQUS, ANSYS, LS-DYNA3D 936, LS-INGRID, LS-NIKE3D, LS-TAURUS, MSC-NASTRAN 69, PDE2D, Spectrum
Visualization and Graphics	AVS, EnSight, FieldView, IDL, NCAR Graphics, TecPlot

Given that much of the government's research funding is for purposes a long way from commercialization, emergence of the commercial software industry is only one measure of the 'success' of government R&D

- Only about 54% (1999) of the total federal R&D spend is on 'development', or research close to commercialization - the rest is not likely to generate any direct commercial return
- Although R&D represents only about 16% of total government procurement (about $198bn in 1996), a large proportion of software simply 'acquired' rather than created for research purposes is 'COTS'
- Hence the 'success' of the government's software purchasing cannot be measured simply in terms of direct contribution to industry - in addition, measures such as the US's intellectual lead, the number of academic papers or citations, and the number of trained human resources must also be considered



A number of mechanisms of transfer to industry of research generated in government labs exist, some more direct than others

- In the late 1970s, the view that government-produced technology (primarily in the GOGO national labs) was not being adequately commercialized gained some support
- The prevailing opinion was that the reason for this was that IP policy was open - since everyone could access the technology, there was little incentive for one company to exploit it since competition could easily follow
- However, believing that this IP approach was fundamentally correct, the government sought to encourage technology transfer by specifically addressing it in legislation, rather than by reducing or changing the IP policy
- Bayh-Dole (1980, PL 96-517), Stevenson-Wydler (1980, PL 96-480), the Cooperative Research Act (1984, PL 98-462), Federal Technology Transfer Act (1986, PL 99-502) were all created with this objective foremost

Mechanisms of technology transfer

- Government sells a system to a private company and then 'rents' back
  - ~ eg ERIM radar
- Publication (eg in professional journals, seminar reports, presentations at professional meetings, and student dissertations and theses)
- Individuals in the 'right place / right time'
  - ~ eg Ray Tomlinson of BBN wrote the world's first email program; he was 'in the thick of the action' through BBN's contracting work building the actual packet links
- Government proactively pushes technology out to industry - eg through the NIST Center for the Utilization of Federal Technology

Many commercial successes have derived from transfer of people rather than specific transfer of technology

- One of the major impacts of federal funding for R&D is to provide a training ground for leading-edge software engineers
- This objective is explicitly addressed in the presidential budget for R&D, and in practical terms the majority of software engineers based in the US working for leading software companies were trained at US universities
- Hence although many of the software projects underway at universities (and other spenders of federal funds) may never see commercial application, the IP is nevertheless transferred indirectly to the private sector
- In addition there are several well known cases in which valuable IP has been transferred along with the individual into the private sector - not always with the blessing of the host institution, however!

Well-known examples of IP transferred to the private sector through individuals

Web browsers
Perhaps the most well-known example - the Mosaic technology was taken in the head of Marc Andreeson and his team to Netscape, where they rewrote the actual code from scratch

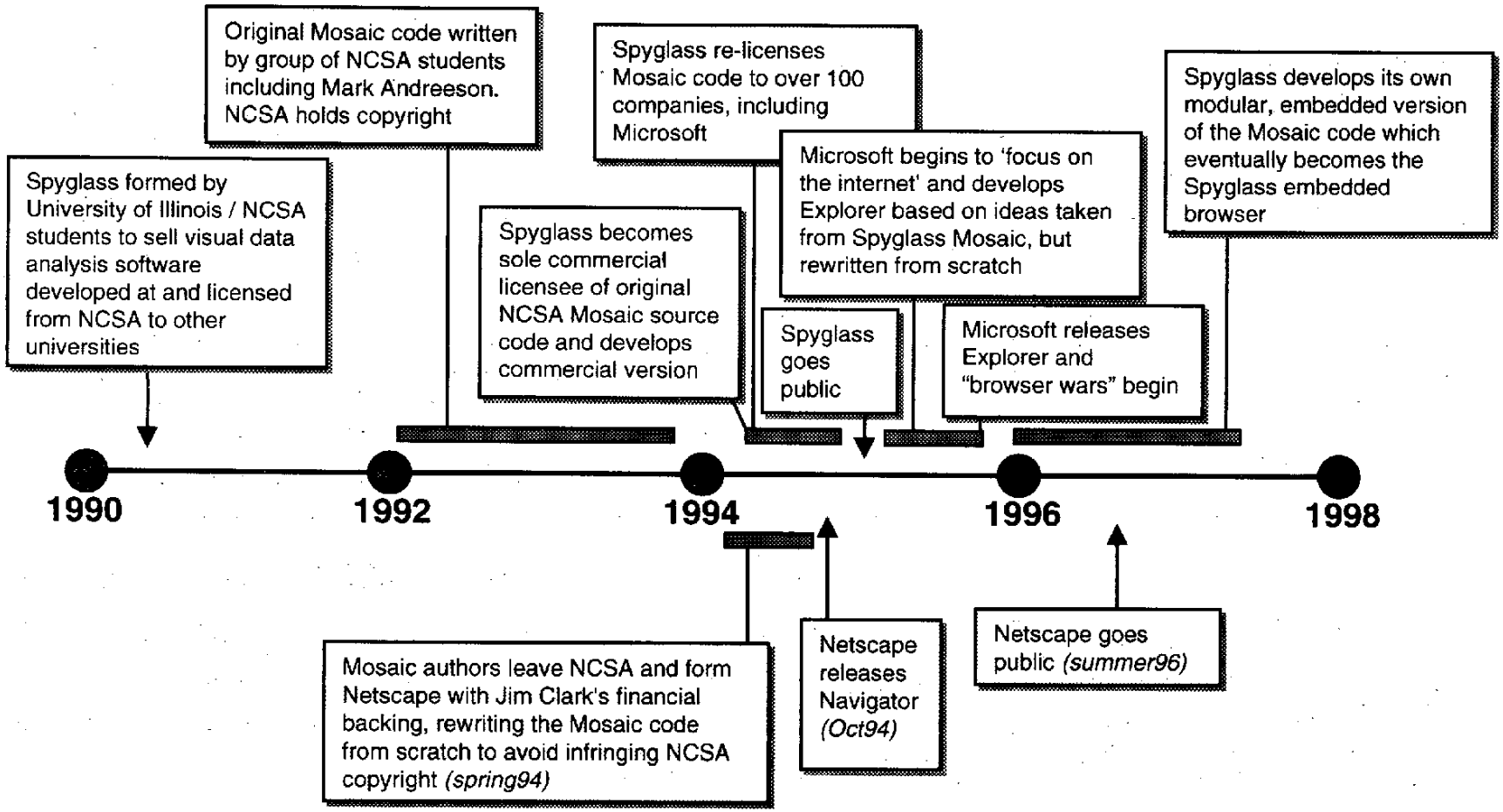
Search engines
Inktomi was the first application for a federally-funded workstation cluster built at UC Berkeley, written by Eric Brewer and Paul Gauthier; David Filo and Jerry Yang developed their first Yahoo catalog and search utilities on Stanford time using Stanford computers (although not working on Stanford projects!); Erik Selberg and Oren Etzioni developed MetaCrawler together at the University of Washington.

Data analysis and prediction
HNC was founded in 1986 to develop 'predictive software solutions' for military applications, and until 1990 operated exclusively using DARPA funds.

Email
Ray Tomlinson wrote the world's first email program while working on BBN's infrastructure project for DARPA

Example - web browsers

The Mosaic / Spyglass / Netscape story illustrates the interplay of IP policy and different transfer mechanisms of IP to industry



Certain research programs currently in existence could potentially have a significant future impact

ATP Focused Program on Component-Based Software	DARPA GLOMO project
• 85% of the installed base of software consists of major custom applications for large customers - expensive, hand-crafted, error prone, and one-of-a-kind • The ATP Program is developing the technologies necessary to enable systematically-reusable software components • Additionally, the Program hopes to encourage an industry around these components - whereby companies would exist to 'trade' the components	• Ad-hoc* networking refers to the concept of networking equipment which automatically finds other network elements and consequently creates a network spontaneously • The DARPA GLOMO project was formed with military applications in mind - the ability to 'throw network equipment out of a plane' and have it form a reliable communications network automatically is clearly appealing
• Program commenced in 1994 • 21 active or completed projects • ATP funding $43.3m • Industry cost share $24.8m	• Commercial applications are very appealing, and beginning to take shape already: - ad-hoc LANs for eg meetings (Spanworks) - rapid roll-out telcos (eg Rooftop, Nomadix) • The IETF has already launched a task force aiming to standardize ad-hoc networking technology

* 'Ad-hoc' is Latin meaning 'do this', and consequently is used to refer to the idea of 'spontaneity' or 'getting on with it'

1 **The US system of federally-funded software IP**

- A Systems of intellectual property protection p6
- B Intellectual property and software p9
- C Federally-funded software IP p20
- D Federally-funded software historical successes p33
- **E The Open Source movement** p41

2 Eight key questions p46

3 Conclusions and implications p58

A *Appendices* p64

- *A1* *List of Sources*
- *A2* *Glossary*

The Open Source movement is gradually gaining strength

Milestones in the history of the 'Open Source Movement'

1969	Unix developed at Bell Labs
1981	'sendmail' is shipped with BSD Unix
1984	Richard Stallman founds the GNU (free Unix) project and the FSF (Free Software Foundation)
1987	First release of Perl
1993	FreeBSD project begins
1991	Linus Torvalds creates Linux
1994	First official version of Linux released
1995	Apache 1 released
1997	'The Cathedral and the Bazaar' by Eric Raymond presented at Linux Kongress
1998 January	Netcraft survey finds Apache used in more than 50% of all websites
1998 February	'Open Source' moniker coined
1998 March	Netscape releases source code for Communicator
1998 May	Corel announces support for Linux
1998 June	IBM announces support for Apache
1998 July	Oracle and Informix announce plans to support Linux; Economist article on Open Source
1998 August	Forbes cover story on Linux and Open Source; Microsoft president Steve Ballmer says 'Sure, we're worried'
1998 September	Intel and Netscape acquire a minority stake in Linux distributor Red Hat
1999 January	HP and SGI announce support for Linux
1999 March	Apple announces plans to open parts of its forthcoming OS upgrade

Open Source is both a symptom and a cause of the gradual shift of parts of the software industry to a service-based business model

- Companies are making money directly from Open Source software by selling service contracts, including for example software upgrades and support
- The same trend is in evidence in commercial software companies - Oracle is hoping to gain 50% of its revenues from 'service' by 2000, and Microsoft is debating the idea of renting its software*
- Many other familiar software applications and even hardware devices are gradually becoming services as network bandwidths increase (see table opposite)
- In a service-based world, there is no particular reason *not* to release the source code - providing the service adds real user value on top of the software itself

Software (hardware) that could be replaced by a service given network bandwidth

- Voice mail / answering machine / PBX
- Audio CD player / library
- Video player / library
- File storage (files could be stored at an online repository accessible from any computer)
- File backup systems (tape, ZIP etc)
- Office productivity suites (Corel is considering a 'rental' or 'pay-per-use' pricing model)
- Streaming video and audio client software

* In the rental model, Microsoft would provide the software and upgrades for a monthly fee rather than a one-time up-front payment. This is in effect indistinguishable from the 'service' model of eg Red Hat. Some would argue, however, that the reason Red Hat can make money from this model when the source code is open is that Linux is not user-friendly enough for anyone to install and use it without support; if Windows were freely available, a much higher proportion of users would be capable of using the code without a service contract from Microsoft

The legal ramifications of Open Source are significant, and if the movement continues to grow are likely to affect government treatment of software IP

- Most Open Source software is protected by some variant of the GPL (GNU Public License) invented by Richard Stallman, which stipulates the terms under which the source code of modified versions must be made available, and specifies that any program which includes other GPL code must itself be made available under GPL
- This effectively maintains the 'spirit' of copyright as it applies to software:
  - a company is free to try to sell the software, but would only succeed in selling to users unaware the software could be obtained freely
  - a company is free to sell the software with added service value on top, and several companies, although small, have proved that this is a viable business proposition
  - a company is free to modify the software, but not to then sell the software as proprietary, closed code - rather, they must use one of the above two models
- Open Source therefore presents a fundamental choice: sell proprietary software, or sell open software plus (proprietary) added value. *If a business can be made around either model, then the second (Open Source) choice is better*, since it contributes more to 'human knowledge', and it may well lead to better software
- Since much government software is in the same category as the software at the center of the Open Source debate (software 'infrastructure' such as operating systems, tools and routines), the government may well embrace Open Source once it believes that a sufficient body of evidence exists to show that viable businesses can be made around Open Source

第２章　８つの問題点への回答

Eight key questions

1	The US system of federally-funded software IP	
A	Systems of intellectual property protection	p6
B	Intellectual property and software	p9
C	Federally-funded software IP	p20
D	Federally-funded software historical successes	p33
E	The Open Source movement	p41
2	**Eight key questions**	p46
3	Conclusions and implications	p58
A	*Appendices*	p64
A1	*List of Sources*	
A2	*Glossary*	

1 Has government's historically open IP policy benefited the US software industry?

2 What is the basic philosophy underlying treatment of the three stakeholders - government, recipients of government funds, and actual (individual) authors?

3 Since software is rarely built from scratch, what in practice does 'software' mean? When new software uses previously existing copyrighted software, what happens to the copyright holder?

4 Is the industry 'open source' movement affecting government software IP policy?

5 How has the US government dealt with the handling of IP generated by multiple parties?

6 In particular for software, how has the government dealt with multiple-party IP?

7 As in Japan, does the US government require source code to be delivered to it?

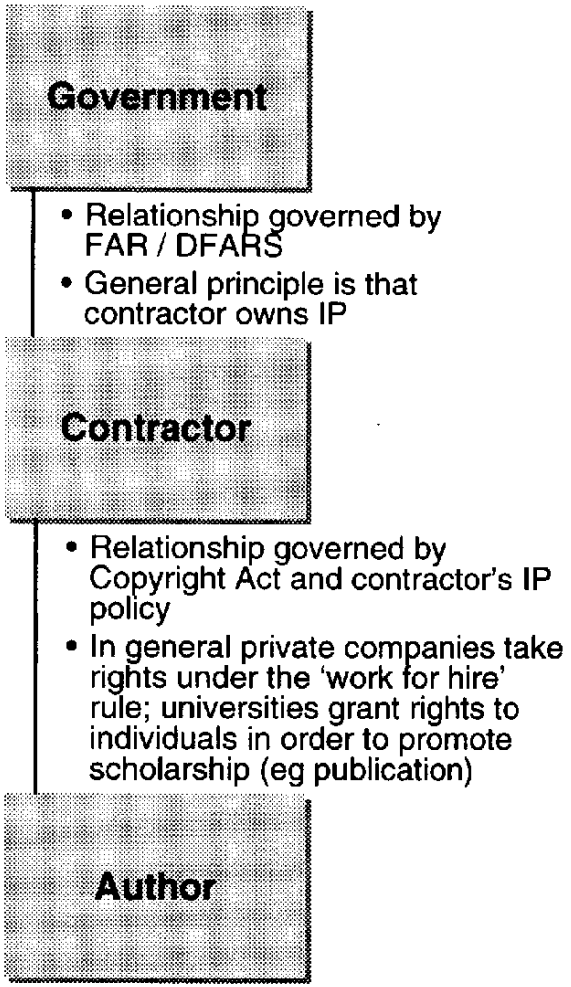
8 Does the US consider software that does not get commercialized a waste of taxpayer funds?

The US software industry has benefited greatly from federally-funded software - but only in part because of the US's open IP policy

- In general, US government IP policy can certainly be said to be open
  - Software developed by government employees or as 'work for hire' is public domain
  - Government generally only obtains limited rights in work developed by contractors
  - Government is generally flexible about not requiring contractors to disclose source code
- Government-funded research has led to numerous benefits for industry
  - Government research has led to the development of many basic algorithms, protocols and sub-components which play a role in industry end-user applications today
  - Given the 'fundamental' nature of much US research (ie its distance from commercial application), the US record in achieving commercial use for its software is quite high
  - In addition, however, government has provided a valuable 'training ground' for many engineers
- Although government IP policy has not always been an active agent in every success story of private sector IP commercialization, however, government has provided a framework under which IP commercialization would not be prevented
  - Many of the research results in use in industry today did not arrive in industry specifically because of government IP policy, but rather through transfer of people
- In addition, since 1980 the government has made active attempts to try to commercialize government-funded technology
  - Bayh-Dole (1980, PL 96-517), Stevenson-Wydler (1980, PL 96-480), the Cooperative Research Act (1984, PL 98-462), Federal Technology Transfer Act (1986, PL 99-502) were all created with this objective foremost

Government policy influences the relationship between government and contractor, but the relationship between contractor and author is generally decided by IP legislation

- Government-generated IP is public domain
- In general, contractor-generated IP is owned by the contractor; however, government generally receives limited rights, which enable government to use the IP for its original purpose, and to use it in future contracts
- The treatment of the actual author depends on the author's contractual relationship with his employer. Typically the author would be either
  - a company employee (in which case the employer owns the IP)
  - a subcontractor (in which case ownership would be determined by contract)
  - an academic researcher (in which case the author would generally own certain rights in the IP, and the university would also retain some rights)



Treatment of IP generated in 'mixed funding' contracts is still a gray area, with many possible scenarios and case-by-case negotiation

- When software developed previously by a contractor and sold commercially is used in developing new software, but in an unmodified form, government generally obtains 'restricted rights', which effectively allow the government to use the software for the original purpose, and to enable future contractors to use it
- Under the DFARS, if the commercial software is actually modified in preparation of the new software, however, government would obtain the same rights in it as it would in the software as a whole (by default, these rights would be 'unlimited rights'); under the FAR, if the commercial software accounts for less than 50% of the total software 'cost', it would receive the same treatment as the software as a whole
- However in practice, when a contractor claims copyright in a work, the government only receives limited rights: under the FAR, the contractor has the right to claim copyright with government approval; under the DFARS, the contractor automatically receives copyright
- In all situations, within this broad framework, the actual IP treatment would be determined on a case-by-case basis. Both the FAR and DFARS contain provisions which enable the specific rights treatment in a particular instance to be modified as appropriate

Government IP policy has not yet been affected by the Open Source movement, although through its recent amendment to the FOIA the government may have unwittingly formed an open source policy!

- Government IP policy does not yet reflect the industry Open Source movement, and no immediate policy changes are being considered
- However, if the Open Source movement continues to gain support, it will almost certainly eventually impact government policy
  - The kind of software developed under government contract overlaps strongly the kind of software being released using the Open Source model
  - Providing a 'business' can be made around Open Source, Open Source is a 'better' model in a large sense, since it contributes more directly to the availability of knowledge, and it can result in better software
  - Again providing a business can be made around Open Source, the IP protection used (the GNU public license) provides a better form of protection than copyright or patents
- Through its 1998 amendment to the FOIA, the government may have instituted a policy by which contractors will have to surrender source code to competitors who ask to see it
- If this were accidental, and government chooses to reverse this policy, then government will have choose between reverting to the more restrictive policy (source code non-disclosure) and potentially moving to a more open, Open Source-like policy

Handling of rights generated by multiple parties is in practice a relatively straightforward issue, with industry generally the winner

- 'Multiple party' relationships generally take two forms: cooperative development agreements (CRADAs), and contractor / subcontractor relationships

- In the case of a CRADA, IP policy is determined on a case-by-case basis, and set out explicitly in the contract. In a CRADA, the government does not directly provide funds to the industry partner (rather, the government's contribution is in the form of federal research personnel time and facilities), and so the contract is not governed by the FAR* and its cumbersome requirements. In practice this means that government has great flexibility to negotiate rights sharing agreements

- In contractor / subcontractor situations, one contractor+ takes the role of 'prime contractor' and is the holder of the contract with the government. It is this contract that determines the rights the contractor will hold, and the rights the government will hold. Separately, the contractor forms normal industry contracts with its subcontractor(s) detailing how rights will be shared. Generally subcontractors retain copyright in their work, passing restricted rights to the prime contractor.

* Instead it is addressed by the Stevenson-Wydler Technology Innovation Act of 1980 as amended by the National Institute of Standards and Technology Authorization Act for fiscal year 1989

\+ In acquisition situations, multiple contractors cannot become prime contractor. In various grant situations (such as the ATP) and CRADAs, multiple parties can form direct contracts with government. In this situation the CRADA approach is used to IP rights management (grants are not covered by the FAR, but by specific Acts)

Software generated by multiple parties follows the same pattern as other forms of multiple-party IP, although occasionally problems can emerge when commercial software is modified by subcontractors

- Software can be generated by multiple parties in three situations: CRADAs, contractor / subcontractor relationships, and contractor purchase of COTS (Commercial Off The Shelf) software
- With CRADAs, software rights distribution are negotiated on a case-by-case basis and generally rights remain with industry
- In a contractor / subcontractor relationship, the rights distribution is as described in answer to question 6. However, occasionally confusion can arise when the rights passed by the subcontractor to the contractor are less than those negotiated by the prime contractor with government. In practice these cases are relatively rare, and are solved by arbitration
- Purchase of COTS software confers restricted rights on its purchaser as a general commercial rule. Even were the contractor to modify the software under special agreement with its original developer, however, the government would still only receive restricted rights (the rules concerning modification of commercial software only affected commercial software modified by its original developer)

The government is generally flexible with regard to delivery of source code, and in many cases does not require its disclosure

- Generally speaking, whatever package of rights the government obtains (unlimited, restricted or government purpose), the government is entitled* to obtain source code, since the government has the right to pass the software to a future contractor, and in practice this can only be done if the government holds the source code
- Under the FOIA, however, the government can be forced to disclose any data in its possession in response to a reasonable request, providing that disclosure would not constitute a threat to national security. Hence if the government received source code, it could be required to pass it to the public
- Since disclosure of software is in many cases tantamount to removal of intellectual property protection, the government generally does not require source code except in cases where there is a certain need for future modification of the source code either by government itself or by future contractors
- Since software is often either a means to an end in the actual research conducted, or intended for its operating purpose only rather than to be modified, the government does not require source code in the majority of cases

* Since neither the FAR nor the DFARS actually use the term source code, however, there is no formal legal basis for entitlement. Instead the entitlement rests on the fact that since in certain situations software is incomplete without source code (such as when the purchaser knows in advance that he will at a later date wish to modify the source code) a contractor could be said to be in default of contract if he did not provide it. However, in practice, establishing this condition would only be possible in situations where the source code was actually necessary to the government, and so the need for the legal definition does not directly arise.

Software funded by government generally pays for itself when it is acquired, so although there may be an opportunity loss if it is not commercialized, there is no waste of taxpayer funds

- Software is funded by government through three mechanisms - contracts (about 60%), grants (about 30%) and cooperative agreements (about 10%) - and under each mechanism, projects can be 'successful' without necessarily achieving commercial success
  - Under a contract, a recipient aims to satisfy a government need or solve a problem in a way directed by government. In this situation, the contractor is conducting a business transaction with the government, and consequently his 'budget' includes an allowance for his profit. Government solves its immediate need, and the contractor makes a profit - consequently even if there is no further commercialization, the transaction is a success. Commercialization of the result is therefore a 'bonus'
  - Under a grant, money is 'granted' to a recipient specifically in order to aid the recipient in some way. An example would be the ATP - government specifically aims to assist the recipient with research leading to a commercial product. In this situation, the project is a failure if a commercial result is not achieved. This would certainly be a waste of taxpayer funds, but more importantly it would mean that the vehicle - eg the ATP - was failing, leading eventually to its abandonment
  - Cooperative agreements are generally formed to conduct research likely to be of benefit both to government (in generating knowledge) and to industry (in leading to commercial profits). IP policy is therefore difficult to mange, but the project could be deemed a success (and a valid use of taxpayer funds) if it were to succeed in meeting either party's needs
- Historically, in the late 1970s, the government was encouraged to achieve a greater return on government-funded technology. Rather than modify IP policy, the government responded with several Acts which provide mechanisms and incentives for increased technology transfer

第3章　日本に示唆するもの

Conclusions and implications

1	The US system of federally-funded software IP	
A	Systems of intellectual property protection	p6
B	Intellectual property and software	p9
C	Federally-funded software IP	p20
D	Federally-funded software historical successes	p33
E	The Open Source movement	p41
2	Eight key questions	p46
3	**Conclusions and implications**	p58
A	*Appendices*	p64
A1	*List of Sources*	
A2	*Glossary*	

The US system of IP management has been successful as a result of a number of interrelated factors

- IP policy with regard to IP developed by contractors is open (in that contractors retain IP) but protective (in that those contractors have the rights to exploit their IP)
- IP developed by federal employees is 'public domain', which has historically acted as a disincentive to commercialization. However realizing this, the government has enacted a number of technology transfer mechanisms and incentives
- The flexibility is provided in the regulations such that complex rights management issues can be handled by case-by-case negotiations, which are held up front before contracts are signed. Consequently if rights issues could not be resolved amicably, generally contracts would not be signed*
- Parallel with government policy, US universities are increasingly being run as businesses managed for a profit; they too are increasingly looking for returns on their IP. Hence universities' proactive approach has been the direct cause of commercial results, but this has been facilitated by government's policy of letting universities keep IP

* Of course, sometimes the most beneficial IP to emerge from a project cannot be foreseen at the start. However, it is in practice rare that multiple parties would have a claim to such 'unexpected IP', and so the original agreement would prevail

The concept of software IP protection is now facing a fundamental challenge, however, which may begin to undermine government's historically open policy

- The software industry is currently facing the so-called 'patent time bomb':
  - The number and reach of software patents issued is rising gradually
  - Since software is not usually 'new', but rather a combination of existing units, the scope for producing software without infringing existing patents is gradually diminishing
  - The response of large companies is to develop huge 'patent libraries' that they can use as weapons for cross-licensing agreements
  - The net effect, however, is divert the time and resources of all companies away from innovation; in time, smaller players who cannot afford the enormous royalty fees they may be required to pay may be excluded entirely

- Software patents could soon begin to impede commercialization possibilities from government-funded development
  - A contractor develops new software for the government, relying on a number of existing patented components, for which royalties must be paid to the owners
  - The contractor receives copyright in the completed work, and believing there is a commercial market, starts to sell the software
  - When sales of the software start to pick up, however, the owner of the original patents writes from scratch software with the same functionality and begins to sell it
  - Copyright is not infringed, because the work was written from scratch. Since the patent holder does not need to pay royalties, his costs are lower, and he can outprice the contractor

Government's response may be to further entrench the historical 'protectionist' approach, but it is also possible government will embrace ideals closer to those of the Open Source movement

- The government will likely act to resolve the software patent issue relatively soon, as industry complaints are growing louder
- The government has several possible policy options, but three in particular stand out:
  - Abolish patents for software
  - Protect federal contractors, perhaps by creating special arrangements such that contractors commercializing federally-funded software are exempt from patent royalties
  - Begin to promote and encourage contractors to make money from service rather than from selling software alone (the Open Source model)
- Of these three main options, the most likely are patent abolition for software and a move of some sort towards the Open Source model
- The determinant will probably be timing - it is still too early to be certain that Open Source is of benefit to industry, and consequently that it would be 'a good thing' to open up government-funded IP. Additionally there is a danger that if 'software as service' companies begin to emerge, they could offer service packages based around open source software developed by others (such as contractors) readily - stealing the market from the actual developers

付属資料

Appendix

1	The US system of federally-funded software IP	
A	Systems of intellectual property protection	p6
B	Intellectual property and software	p9
C	Federally-funded software IP	p20
D	Federally-funded software historical successes	p33
E	The Open Source movement	p41
2	Eight key questions	p46
3	Conclusions and implications	p58
A	***Appendices***	p64
A1	*List of Sources*	
A2	*Glossary*	

Comparative Studies in Software Acquisition, Steven Glaseman, 1988

Owning The Future, Seth Shulman, 1999 Houghton Mifflin

Protection of US Competitiveness in the International Software Markets, James P. Chandler, George Washington Journal of International Law and Economics, 1991

Intellectual Property and the National Information Infrastructure, Working Group on IP Rights, Information Infrastructure Task Force, 1995

Legally Speaking: The NII IP Report, Pamela Samuelson, Law Professor, UC Berkeley, 1995

Technical Data and Computer Software: A Guide to Rights and Responsibilities under Federal Contracts, Grants and Cooperative Agreements, Council on Governmental Relations, 1996

Intellectual Property Protection for Software, Carnegie Mellon University Software Engineering Institute, 1989

Intellectual Property and Software: The Assumptions are Broken, Randall Davis, MIT AI Lab, 1991

The Challenge of Software-related Patents, David R. Syrowik, Software Patent Institute, 1994

Selected Issues in Apportioning Data Rights in Software Development Contracts, James D. Gibbs, Gibson, Dunn & Crutcher, 1992

Distributing Risks on Software Development Contracts, Michael Mutek, Hughes Information Technology Company, 1992

Software Development: A Legal Guide, Stephen Fishman, 1998

Patent, Copyright and Trademark, Stephen Elias, 1999

Open Source: The Emergence of a New Model, O'Reilly & Associates, 1998

COTS	Commercial Off The Shelf
CRADA	Cooperative Research and Development Agreement
DFARS	Defense Federal Acquisition Regulations Supplement
FAR	Federal Acquisition Regulation
FOIA	Freedom of Information Act
IP	Intellectual Property
NDA	Non-Disclosure Agreement
OFFP	Office of Federal Procurement Policy

米国の政府支援研究開発プロジェクトにおける
知的財産権の取り扱いの変遷の歴史とその背景

© 平成11年3月発行

発行所　財団法人 日本情報処理開発協会
先端情報技術研究所
東京都港区芝2丁目3番3号
芝東京海上ビルディング4階
TEL（03）3456-2511