A Standard's Integrity: Can It Be Safeguarded?

Tineke M. Egyedi, Delft University of Technology
John Hudson, Bath University

This article is partly based on a research project funded by the European Commission (DG Enterprise, Standardization Unit), funding for which we are very grateful. The Commission need not necessarily agree with the conclusions drawn. We further thank the participants of the EURAS 2001 workshop, the anonymous reviewers, and Mostafa Hashem Sherif for their comments.

[1] A de facto standard is a specification that is not officially accredited by any standards body, such as the International Organization for Standardization (ISO), consortium, or forum (e.g., ECMA, IETF, and IEEE). It arises out of market dominance of one product. An example is the IBM compatible PC standard, which became the norm for PCs in the 1980s.

[2] It must be stressed that in the wordings of [15] the research question focuses on "compatibility standards" and not on "similarity standards" like those for the voice coder or video coder. The latter can only be implemented in full or not at all (i.e., they cannot be adapted, extended, or selectively implemented).

ABSTRACT

When a compatibility standard is adapted, extended, or selectively implemented, its purpose is likely to be undermined. Its value declines sharply because interoperability between standard-compliant implementations becomes uncertain. The integrity of the standard is at stake. This article discusses strategies intended to safeguard standards' integrity. The examples are set in the context of the development of Java, a de facto standard. In what manner and to what degree can the integrity of standards be protected?

INTRODUCTION

Irrespective of whether they are committee standards or de facto standards[1] of proprietary or open source origin, standards integrity is important. It is important to companies who have invested in standard-compliant research and development (R&D) and their customers. When an implementation deviates from a standard, this fact is usually not made public for it reduces the standard's network effects. That is, it diminishes the product's interoperability with products of other companies and may fragment the market. When deviations are introduced, the interoperability of standard-conforming products is not self-evident anymore. There may be good reasons for introducing changes to a standard when implementing it. For example, some features may be superfluous for the intended context of use. There may also be "bad" reasons. For example, introduced deviations may be a strategy to frustrate the development of a competitive technology and lock customers into a proprietary technology. However, regardless of the intention, the outcome can be detrimental to market and technology development.

Chadwick's analysis [1] of Microsoft's use of Kerberos in Windows 2000 illustrates the problem of deviant standards. With Windows 2000, Microsoft replaced its proprietary Windows Internet Naming System (WINS) and supported open Internet standards for the Domain Name System (DNS) service (RFC 2136 and RFC 2782). However, "… dynamic updates to a DNS service leave you open to security breaches via the Internet" [1]. Microsoft took a different Internet standard, Kerberos v5 (RFC 1510), as a starting point to address this issue but altered it in a way that prevented proper interoperability with other Kerberos v5 implementations. The result was that "… if you use Microsoft's tweaked version of Kerberos v5, you can verify non-Microsoft clients, but if you're using your own server security, you can't verify a Microsoft client" [1, p. 107].

There are several undermining strategies that cause interoperability problems and require cumbersome, extra effort by third parties to repair. A well-known one is to take a standard as the starting point and elaborate on it by adding extra functionalities (*embrace-and-extend* strategy). We draw attention to two other strategies as well: not implementing part of the standard (*embrace-and-omit* strategy) and introducing local adaptations to standards (*embrace-and-adapt* strategy).

In this article we focus on issues that are at stake *when (de facto) compatibility standards are adapted, extended, or selectively implemented*.[2] We refer to these instances as problems of integrity — that is, a specific subset of compatibility problems. If the integrity of a standard is harmed, fragmentation of the market looms. This affects the wider business community as well as the general public. An incompatible fragmented market is a problem of public interest (i.e., a problem in respect to fair competition, technological innovation, and market transparency [3]). Are there mechanisms that safeguard the integrity of standards?

In the following, we explore to what degree, in what manner, and in which settings the public interest in the integrity of (de facto) standards may be safeguarded. We focus on mechanisms of governance and control:

- The committee setting in standards' development; that is, the committees of formal standards' bodies, accredited standards' organizations, and standards' consortia
- The market setting in standards' development, where specifications can develop into de facto standards and standards' implementation; that is, the market as it affects the way standards are used in practice
- The legislative setting, which is the *default regime* when the coordinative mechanisms of both the standards and market settings fail [4]

Because there is little literature to draw on, McGowan and Lemley [3] excepted (see further

In January 2000 a higher court confirmed that unfair competition was at stake, not copyright infringement. In January 2001 the dispute was finally settled. Among other things, the TLDA was terminated and Microsoft agreed not to use Sun's Java Compatible logo in the future.

on), we follow a case-based inductive approach. We seek to answer these questions by examining the difficulties relating to the integrity of the Java platform (Java) in the late 1990s and early 2000s.

THE JAVA CASE

Java started as a programming language under the auspices of Sun Microsystems (Sun). One of Sun's maxims is "Write Once Run Anywhere" (WORA). That is, a Java programmer should not need to rewrite his or her software to make it run on the proprietary platforms of different vendors. In order to achieve cross-platform compatibility, Sun and other Java programmers defined a standardized programming environment.[3] To achieve WORA, each system and browser provider had to fully implement this environment.

Java became popular because small — platform-independent — Java programs could be downloaded and executed by Web browsers. These innovative, moving, colorful applets triggered Java's breakthrough on the Internet. Java was already a de facto standard when, in 1997, Sun approached Joint Technical Committee 1 (JTC1) of the International Organization for Standardization (ISO) and the International Electrotechnical Committee (IEC) to formally standardize it. Sun took the first step in the formalization process, and became a recognized submitter of publicly available specifications (PAS) in late 1997. This status allowed the company to use a procedural shortcut through formal standardization. However, Sun refrained from pursuing this option because it feared that its Java specification would be changed — instead of ratified — during the approval procedure. Sun withdrew from JTC1 and approached ECMA International, an international industry association for standardizing information and communication systems, in April 1999. Again, it did so to formalize Java's de facto status. Once an ECMA standard, Java could more easily be ratified by JTC1. An ECMA standards committee was formed. However, after the first committee meeting Sun withdrew. ECMA's Intellectual Property Right (IPR) rules were not elaborate enough, according to Sun. Instead, Sun preferred to redesign, open up, and elaborate the Java Community Process (JCP) to further develop and maintain Java.

Because it lessens dependence on the Windows platform, Java's promise of platform-independent computing made Microsoft nervous as early as 1995 (*Findings of Fact* antitrust lawsuit against Microsoft). Microsoft urged companies such as Netscape and Intel to withdraw from activities that supported Java developments. In March 1996 Microsoft appeared to give in. Sun and Microsoft signed a *Technology License and Distribution Agreement* (TLDA) for the use of Java. Part of the agreement was that Microsoft would incorporate Java in its Internet Explorer (IE) 4.0. However, in order to maximize IE's use *vis-à-vis* Netscape Navigator, Microsoft more tightly integrated version 4.0 with the Windows platform. Moreover, the company introduced Java development tools for the Windows platform and developed an incompatible Java programming environment, both of which undermined WORA. A lawsuit ensued. Summarizing the highlights, in October 1997 Sun filed a complaint against Microsoft for copyright infringement. In March 1998 the court granted Sun's request for a preliminary injunction: Microsoft was not allowed to use the *Java Compatible* logo unless its products passed Sun's test suites. In May Sun filed an additional complaint for unfair competition. In November 1998 the court ordered Microsoft to change its Java software and development tools. Microsoft appealed against the ruling. It argued that the punishment did not fit the crime. It pleaded guilty on "breach of contract," for which the TLDA stipulated a much lighter punishment than injunction. In August 1999 the court granted Microsoft's appeal. Sun protested. In January 2000 a higher court confirmed that unfair competition was at stake, not copyright infringement. In January 2001 the dispute was finally settled. Among other things, the TLDA was terminated and Microsoft agreed not to use Sun's Java Compatible logo in the future.

LESSONS FROM THE JAVA EXPERIENCE

From an economic perspective, standardization of goods and services has two main purposes: to reduce transaction costs and to achieve economies of scale through product interchangeability (or *interoperability*) [5]. Standards reduce transactions costs because both parties to a deal mutually recognize what is being dealt in [6, p. 378]. Given the complexity and uncertainty associated with the consumption of many goods and services, it is not unusual for consumers to find themselves at an information disadvantage relative to producers. To reduce the information asymmetry, consumers, in this case including other firms, gather knowledge of the service or product they are buying from either past experience, word of mouth, seller "signaling," or whatever [7, p.1342]. Standards also contain such information. Firms that purchase goods may minimize their transaction costs by restricting their search for suppliers to those whose products comply with compatibility standards. Thus, standards-conforming products reduce transaction costs and locally restore informational symmetry between producers and consumers. However, this only applies if the integrity of the (de facto) standard is maintained. In the interest of other market players, suppliers should be clear about the features and functionality of their products and services. If they advertise that their products conform to a standard, consumers must be certain that this claim is fully met, all relevant information has been provided, and compatibility and interoperability will be achieved. This applies to formal standards as well as de facto standards like Java.

The case highlights three kinds of integrity control (Table 1). First, the integrity of a standard (i.e., the Java platform) may be safeguarded by keeping the technology under proprietary control, for example, by licensing it under protective conditions and exerting ownership rights if standards integrity is threatened. In this option the market is the setting of governance (e.g., the Sun-led JCP). The second option is that a standards committee becomes custodian of the specification. This could be either a formal standards

[3] The standardized application programming environment includes the specifications of the Java Virtual Machine (JVM; i.e., software that runs on proprietary operating systems and is capable of interpreting compiled Java byte code) and application programming interfaces (APIs). APIs comprise the standard packages, classes, methods, and fields made available to software developers to write programs.

Kind of integrity control	Governance setting	Assigns custodianship to (e.g.)	Main interest at stake (nature of interest)
(1) Proprietary	Market	Market tools (e.g., license)	Company interest (ownership)
(2) Democratic	Standardization	Standards committee procedures	Multiparty interest/ interest of dominant actor in the network (market/compatibility)
(3) Legislative	Legislation	Legislation and the justice system	Public interest (transparent market, fair competition, innovation)

Table 1. *Overview of types of, settings of, and tools for integrity control, and the main interests at stake.*

committee or a committee of another standards developing organization (i.e., JTC1 or ECMA). In this option, allegedly democratic committee procedures govern standards maintenance. The rationale behind committee control is, primarily, that committee members have a shared interest in maintaining the integrity of the specification because of the economies of scale that result. The third option is the legislative one. It takes a public interest angle on standards' integrity. McGowan and Lemley [3], who take a legal perspective on Java platform independence in the Microsoft antitrust case, discern three types of public interest in compatibility: a transparent market (which reduces transition costs), fair competition, and technological innovation. If one of these interests is harmed and coordinative governance mechanisms of standards committees and the market fail, legislative action can be invoked to enforce standards integrity.

In the following we examine to what degree the three options are effective in safeguarding standards' integrity. We first discuss voluntary control mechanisms by means of standardization and the market. Next, we explore whether the legal setting provides instruments that protect standards' integrity interests. We will then be in a position to reach conclusions and make some policy recommendations.

Is There a Voluntary Way to Protect Standards Integrity?

The literature on de facto standards has mainly focused on their evolution and effect on the market structure [8, 9]. This has shown that if the process of coordinating technology development is left to the market alone, the outcome may be unsatisfactory in several respects. Under conditions of positive network externalities, *bandwagon phenomena*[4] occur that may result in a market's premature commitment to a de facto standard of inferior technology. Lock-in effects occur where (de facto) compatibility has developed as a result of market dominance, and dominance in itself has become a positive attribute for buyers. Moreover, if one or more firms have control of a de facto standard, the problem of anti-competitive behavior arises. These firms will be able to internalize differentially higher costs on their rivals, and even deny them market access [11].

Farrell and Saloner [12] compare the effectiveness of committees and markets in creating compatibility where incompatibility exists or is expected. Their reasoning starts with an assumption of a need for coordination. Standards committees are an explicit response to this need. The bandwagon mechanism is an implicit response by the marketplace. But the market mechanism does not always achieve standardization. The bandwagon mechanism is imperfect when there is no clear leader and when there are different preferences among standards. However, committees too are imperfect coordinators. Often, by the time a committee is convened, participants have vested interests in incompatible positions. [12, pp. 236–37]

Our question addresses the subsequent stage of a standard's development. If coordination has already been achieved by means of a (de facto) standard, which of the two environments is best equipped to prevent different versions of the standard developing? The literature on committee processes still has some relevance here, because if a committee is to safeguard a standard's integrity, this will, of course, also be a coordinated effort. The mechanisms that underlie collective decision making [10] still apply; compromises will still result if interests diverge; and committees, open to capture by large firms and coalitions of producers during standards development [11], may also be captured during the standards maintenance stage.

In situations where committees are imperfect coordinators with regard to standards development, they are also imperfect with respect to safeguarding standards' integrity. And the same can be said about markets. The above literature on (de facto) standards development contains strong arguments against the market and proprietary control of (de facto) standards' integrity. But there the comparison ends. Standards development and maintaining standards integrity are characterized by different processes, and hence there is a need to address different problems, too. (For example, the bandwagon mechanism plays no obvious role in this area.) We will let the Java case speak for itself, and illustrate the problems of standards' integrity that played a role when Sun withdrew from the JTC1 PAS process. See Box 1.

Box 1 mentions several market-, standardization-, and legislation-oriented initiatives that could be interpreted as a means to protect the integrity of the Java platform. They are listed in Table 2. Standards integrity is not a temporary or time-specific issue. It is a matter of continuous concern. Some of the measures listed in Table 2 require continuous vigilance and would therefore seem more ineffective than others. For example, Sun's problem with JTC1 and ECMA standardization was that, although it could determine the starting point of standardization (full

The bandwagon mechanism is imperfect when there is no clear leader and when there are different preferences among standards. However, committees too are imperfect coordinators. Often, by the time a committee is convened, participants have vested interests in incompatible positions.

[4] *Farrell and Saloner [11, p. 236] define the "bandwagon mechanism" as: "If an important agent makes a unilateral public commitment to one standard, others then know that if they follow that lead, they will be compatible at least with the first mover, and plausibly also with later movers."*

When Sun approached JTC1 to formalize Java, one of its goals was to "preserve industry's substantial investment in Java." This was a way of saying that Java should not undergo serious changes during the PAS review process. Sun further expected to retain control over the standards maintenance process by securing the role of the Java community during JTC1 standardization, whose input was coordinated by Sun itself. Sun further upheld its essential IPRs, and retained its patents (although no fees were asked), its copyright (joint copyright ownership was suggested, no fees asked), and trademarks (e.g., control over the compatibility logo). The revenues from IPRs were forfeited in exchange for enlarging and stabilizing the Java market — without compromising control over cross-platform compatibility (ensured, e.g., by means of the Java compatible logo and the test suites). JTC1's role was to codify and ratify the specification development activities supervised by Sun.

Sun withdrew from the PAS process because the PAS procedure was changed. The new procedures, according to Sun, implied that Sun would have had to turn over standards maintenance and control over the evolution of Java to JTC1. Moreover, standards maintenance would not be restricted to minor adjustments such as bug fixing. The changes signaled that Sun would encounter problems when submitting the Java specification. For example, a Java Study Group had been installed in JTC1 (SC22), and people were discussing how they were going to change the Java specification. In addition, there were market developments that threatened Sun's position and increased its desire to keep a grip on Java developments. First, Microsoft did not abide by the Java licensing agreement and posed a threat to cross-platform compatibility. The lawsuit discussed earlier followed.

Second, there were unsettling developments in the area of real-time embedded Java. In 1998 workshops were organized to develop specification requirements for real-time Java. This led to the formation of the Real-Time Java Working Group (RTJWG) led by Microsoft and Hewlett-Packard. The RTJWG activities were disquieting to Sun, because real-time Java draws on the base specifications of Sun's Java. The working group could not write real-time specs in a useful way without making changes to the base specifications. There was thus a risk that competitive developments in the field of real-time Java would affect the work done on Java within Sun's JCP and hence endanger the integrity of the Java platform.
NB: Shortly afterward, Sun carried through changes to the Java platform to increase its usability for the real-time market [13].

Box 1. *Sun's withdrawal from JTC1: integrity issues.*

Integrity control / *Sun initiatives*	Input control	Output control
Standards development/ratification in JTC1/ECMA	High	Low
Standards maintenance procedures in JTC1/ECMA	Medium	Low
Java Community Process procedures	High	Medium–high
Licenses (TLDA/Sun Community Source)	—	High
IPRs	—	High
Test suites and compatibility logo	—	High

Table 2. *An assessment of the degree of integrity control achieved by Sun's initiatives regarding Java (1996–2000) [14].*

control of the initial committee input, namely the Java specifications, and a large say in the procedures to be followed), it could not fully control what would happen to Java during the standards maintenance phase. That is, its input control was high, but its grip on platform integrity during the protracted maintenance phase (i.e., its outcome control) would have been very limited. Other measures such as licenses and test suites seem better equipped to protect standards integrity *because* they are specifically invoked for the purpose of output control (Table 2).

A strong, persistent incentive is needed to secure industry's sustained voluntary compliance to standards. Analogous to the notions of "market push" and "market pull," we speak of *compatibility push* and *compatibility pull*. For example, the Java compatibility logo and the benefits expected from carrying the logo created a strong incentive for other players to develop compatible commercial products (i.e., compatibility pull). Sun also used the complementary approach to increase standard integrity, for example, by encouraging the use of a Java toolkit, SDK (i.e. compatibility push). Use of the same toolkit is less likely to result in incompatible implementations.

Such mechanisms of coordination can be coercive but, like standards' committees, they do not enforce standards' compliance (i.e., little output control). It would seem that only standards prescribed by government provide the legal pressure not to deviate from the committee standard (i.e., compatibility push). However, currently, such *de jure* contexts are the exception rather than the rule in the field of IT and for compatibility standards in particular, and any change in this respect is not likely in the foreseeable future.

Does Legislation Safeguard the Integrity of Standards?

Since Schumpeter there has been considerable interest in the need to reward innovation with some form of market protection such as patents or copyright. In general, it is argued that society must equate the gains from rapid technological progress against the welfare costs of generating temporary monopolies (e.g., [15]). In policy terms, a trade-off must be made between encouraging innovation (which is thought to be stimulated by IPRs), minimizing the harmful effects of the possible monopoly abuse that may result, and encouraging competition. In this context, safeguards for maintaining compatibility are a means to counter anticompetitive consequences of IPRs. Are they used for this purpose? Where committee and market processes fail, will the default regime of legislation protect the public interest in integrity? Does legislation acknowledge the importance of compatibility — standards integrity issues included — in its own right?

Compatibility interests do not have a firm law-based status. They are referred to in relation to patent protection and copyright issues in European, U.S., and international regulation, but only in a marginal way.[5] A legal clause in European and U.S. copyright law allows copyright infringement for the sake of interoperability (i.e., interface information for achieving interoperability with an independently created program and reverse engineering for the purpose of interoperability, respectively). The proposed European patent directive includes the same restricted exception. U.S. patent law does not [4 pp. 111–13]. In some countries copyright law includes the moral "right of integrity" of a work, which aims to prevent others from distorting the work. Possibly this right could also apply to the compatibility implications of standards.

Sun's legal dispute with Microsoft would seem to confirm the lack of a law-based status for compatibility interests. Briefly recapitulating, Sun accused Microsoft of intentionally fragmenting the Java platform by abusing Java's name and compatibility logo. It sued Microsoft for copyright infringement — and, at a later stage, for unfair competition. The court ruled that only the "unfair competition" part could be sustained. The ruling suggests that compatibility arguments are not well institutionalized in U.S. law. They need to be worded in terms of "competitive market interests" to be acknowledged. More generally, the political and legislative system places relatively little value on compatibility as a public interest issue.

Discussion

We began with the question of how to best administer the integrity of de facto and committee standards once they have emerged. Can private companies, standard committees, or legislative instruments be expected to administer standards optimally? As far as the Java de facto standard is concerned, the lesson is a hard one: the integrity of standards cannot be protected by current means. We could refer to it as the problem of the "commons of compatibility." Standards' integrity is not likely to be safeguarded by the communal arrangements of standards committees. Furthermore, the concepts of standards' integrity and compatibility interests are virtually absent in law and regulation. A standard's integrity can only be safeguarded to some degree in situations where compatibility interests can be understood as (translated into) straightforward economic interests (e.g., property interests). This would seem a very meager conception of the "public interest."

The lack of attention standards integrity and the protection of compatibility interests receive is evident beyond de facto standards. It also applies to committee standards. There are three vehicles: governance, market committee, and legislation. Each was found to be at fault when it comes to maintaining standards' integrity. Should they be reformed? Should the courts be more ready to recognize infringement of a standard as a legal issue, also in respect to committee standards? This is a political decision, of course, which requires further research. Is it possible to engender a market solution to the governance of standards integrity that is not subject to monopoly bias? Can committees be reformed so that there is a special procedure for recognizing de facto standards? We believe that progress can be made on all of these issues, and that in particular the law as the default or fallback regime should be strengthened in this area. However, the starting point for this policy reform must be a fully informed debate on the issues and implications, and the practicalities of the solutions.

We believe that progress can be made on all of these issues, and that in particular the law as the default or fall back regime should be strengthened in this area. However, the starting point for this policy reform must be a fully informed debate on the issues and the implications, and the practicalities of the solutions.

References

[1] D. Chadwick, "Windows 2000: A Threat to Internet Diversity and Open Standards?," *Computer*, Aug. 2000, pp.107–09.

[2] M. H. Sherif, "A Framework for Standardization in Telecommunications and Information Technology," *IEEE Commun. Mag.*, no.4, Apr. 2001, pp. 94–100.

[3] D. McGowan and M. A. Lemley, "Could Java Change Everything? The Competitive Propriety of a Proprietary Standard," *Antitrust Bulletin*, vol. 43, 1998, pp. 715.–73.

[4] T. M. Egyedi, "IPR Paralysis in Standardization: Is Regulatory Symmetry Desirable?" *IEEE Commun. Mag.*, vol. 39, no.4, 2001, pp. 108–14.

[5] T. M. Egyedi and J. Hudson, "Maintaining the Integrity of Standards: No Passage East of Java," *EURAS Yearbook of Standardization*, forthcoming.

[6] C. P. Kindleberger, "Standards as Public, Collective and Private Goods," *Kyklos*, vol. 36, 1983, pp. 377–96.

[7] H. E. Leland, "Quacks Lemons and Licensing: A Theory of Minimum Quality Standards," *J. Political Economy*, vol. 87, no. 6, 1979, pp. 1328–46.

[8] P. A. David, "Clio and the Economics of QWERTY," *Economic History*, vol. 75, no. 2, 1985, pp. 332–37.

[9] J. Farrell and G. Saloner, "Standardization, Compatibility and Innovation," *RAND J. Economics*, vol. 16, 1985, pp. 70–83.

[10] K. Arrow, "Alternative Approaches to the Theory of Choice in Risk-Taking Situations," *Econometrica*, vol. 17, 1951, pp. 404–37.

[11] P. A David and W. E. Steinmueller, "Economics of Compatibility Standards and Competition in Telecommunications Networks," *Info. Economics and Policy*, vol. 6, 1994, pp. 317–41.

[12] J. Farrell and G. Saloner, "Co-ordination through Committees and Markets," *RAND J. Economics*, vol. 19, no. 2, 1988, pp. 235–52.

[13] T. M. Egyedi, "Why JavaTM was -Not- Standardized Twice," *Comp. Stds. & Interfaces*, vol. 23, no. 4, 2001, pp. 253–65.

[14] T. M. Egyedi, "Strategies for De Facto Compatibility: Standardization, Proprietary and Open Source Approaches to Java," *Knowledge, Technology, and Policy*, vol. 14, no. 2, 2001, pp.113–28.

[15] G. C. Loury, "Market Structure and Innovation," *Quarterly J. Economics*, vol. 93, 1979, pp. 395–410.

Biographies

TINEKE M. EGYEDI (T.M.Egyedi@tbm.tudelft.nl) is senior researcher on information and communication technology (ICT) standardization at Delft University of Technology, the Netherlands. She recently chaired the 3rd IEEE Conference on Standardization and Innovation in IT. She is a member of the editorial board of two international journals on ICT standardization (*CSI* and *JITSR*), and vice president of the European Academy for Standardization. She has published widely and guest edited several special journal issues.

JOHN HUDSON (J.R.Hudson@bath.ac.uk) is a professor of economics at the University of Bath. He has published three books and more than 60 articles in leading journals such as the *Economic Journal*, *Journal of Economic Perspectives*, *European Journal of Political Economy*, and *Journal of Public Economics*. He has also advised the tax authorities in the United Kingdom and the South West Regional Development Agency. His work covers all areas of economics including industrial economics, standardization, development economics, and the economics of voting behavior.

[5] *For example, the World Trade Organization (WTO) Agreement on Trade-Related Aspects of Intellectual Property Rights (TRIPS) of 1994; European Council Directive 91/250/EEC of 14 May 1991 on the legal protection of computer programs; proposed Directive on software patents (COM 2002/92), Official Journal of the European Union C61/154-163 of 14.3.2003; Digital Millennium Copyright Act of October 1998.*