

The UCITA Amendments: What Do They Mean?

Cem Kaner¹

In the summer of 2001, the American Bar Association (ABA) appointed a working group to study the Uniform Computer Information Transactions Act (UCITA). The author of UCITA is the National Conference of Commissioners on Uniform State Laws (NCCUSL). A NCCUSL committee (UCITA standby committee) meets as needed to suggest revisions to UCITA. They met in November, 2001 to consider proposed amendments to UCITA. The meeting was attended by the ABA working group as non-participating observers.

Dozens of amendments were proposed, including thirty from the Americans for Fair Electronic Commerce Transactions (AFFECT) (a coalition of consumers, librarians, retailers, non-software manufacturing businesses, financial institutions, and software development professionals).²

The NCCUSL working group issued a set of recommended amendments to UCITA in December.³

The ABA working group issued its report in January.⁴ The report started from the assumption that the December amendments would be adopted by NCCUSL⁵, raised additional concerns and recommended a further 19 amendments.

The UCITA standby committee issued a final set of recommended amendments, with commentary in May 2002.⁶

In the summer of 2002, NCCUSL adopted 38 amendments to UCITA,⁷ which were summarized recently for readers of UCC Bulletin.⁸

The amendments clarify and improve UCITA in several ways, but fundamental problems remain. AFFECT has provided an amendment-by-amendment analysis⁹ that reflects the diversity of interests in the coalition. My interests, and this paper's analysis, are primarily focused on the impact of UCITA on the mass-market customer and on the software development community.

CONTEXT: THE UCITA CONTRACT

UCITA's opponents (and defenders) often focus discussion on UCITA's individual sections, attacking or justifying them one by one. The vehemence of the opposition doesn't come as a reaction to the individual issues, but to the gestalt, the collective effect.

A customer decides to buy a software product of a certain type, such as a word processor, a computer game, or a programming language.

- On attempting to do comparison shopping among products within the category, the customer discovers that none of the leading vendors make their contract terms, including warranty and customer-support terms, available before the sale. The customer contacts the software publisher directly, requesting a pre-sale copy of the contract but is politely rebuffed. To see the contract, the customer will have to pay for the product and load it onto her computer.
- The customer is also unable to find trustworthy information about the defects in these products. Software publishers routinely ship products with hundreds or thousands of known defects¹⁰, so the question is not whether the product will be flawed (it will be) but whether the specific flaws will interfere with this particular customer's intended uses. The publishers do not provide information about known defects in their products and the customer finds it difficult to distinguish accurate descriptions from error and disinformation in the myriad of web sites that post product discussions.

- The customer looks for comparative reviews of the products in magazines, but detailed factual comparisons are unavailable because of restrictions in the software license on publishing product reviews or the results of benchmark studies.
- Eventually, the customer picks a product and begins to install it on her computer. At this point, she can read the contract by scrolling through a long license contract displayed in a very small box on the screen.
- While reviewing the contract, the customer has a “right of return.” If she objects to the terms, she can return it to the vendor who sold it to her.
- The customer cannot use the product until she clicks “OK” to “accept” the contract. As soon as she does this, before she has any opportunity to see how (and whether) the program actually works, her right of return expires.
- The contract terms are non-negotiable. Knowing that few customers will read the contract at this point, and of those who do, few will actually reject the product in hopes of finding (through this process of trial and error) another with better terms, the vendor includes several harsh terms.
- As examples of the terms, all implied warranties are disclaimed. Remedies are limited. The license excludes reimbursement for incidental and consequential damages, and provides as the sole remedy, replacement of the disk on which the software is stored. The limitation is made specifically applicable to losses caused by viruses or by defects known by the vendor and not revealed to the customer. The product is not transferable--when the customer's daughter is finished playing the computer game, she cannot put everything back in the package and give it to the boy next door. The product may be used for one year, after which it will stop working unless the customer pays a new fee. The product may be used only on the original machine on which it was loaded. To transfer the software to a different machine owned by the customer, the customer must phone the publisher for permission and pay a transfer. The customer or her agent may not reverse engineer the product to discover or repair defects or security holes. Details of the operation of the product are trade secrets of the publisher and by accepting this license, the customer agrees to protect the confidentiality of this information. In particular, the customer may not publish descriptions of the details of the operation of the product, including the results of benchmark studies, without the specific permission of the company. Customer usage of the product will be tracked by the software and uploaded to the vendor when the customer connects to the Internet. The vendor states that this is to be used to improve the product or to assist the vendor in tailoring the product for that particular customer, but the details--what information is being collected and how it will be used--are not disclosed to the customer. All claims arising out of or in connection with this agreement are to be resolved by arbitration, using the <vendor's choice> arbitration organization. Notwithstanding the arbitration clause, the vendor shall have the right to enforce its intellectual property rights in federal court, in the District of <vendor's choice>.¹¹
- Customers are welcome to call the vendor for technical support, but they must pay \$5 per minute of support time, even if the call arises out of a defect in the product or an error or omission in the user documentation.

This is a fundamentally one-sided contracting process, substantially different from what we normally expect in sales of goods (UCC Article 2, supplemented by various consumer protection laws that apply to goods).

UCITA was drafted as a modification of UCC Article 2, crafting new rules that would be more applicable to software (and later, information) transactions. Many different stakeholders are

involved in these transactions. Opposition comes from stakeholder groups who believe that they have lost much more than they've gained from the new rules.

SPECIFIC CONCERNS

1. *Nondisclosure of Terms*

The ABA working group concluded that UCITA should require the software licensor to “make all the terms of the license transaction available for review by the licensee before the licensee pays, or becomes obligated to pay” for the software, pointing out that it is very easy for the licensor to post the terms on its own website.¹²

The UCITA standby committee responded that requiring vendors to disclose these terms before the sale “would impose substantial new distribution and record-keeping costs and risks”, and provided several examples.¹³

The argument might have been compelling if we did not already have the Magnuson-Moss Act, 15 U.S.C. section 2301, for goods. Makers of consumer goods have been supplying contract terms to customers who want to see them, whether they are in-store shoppers, mail-order-catalog customers, telephone customers or internet customers, since 1975. Indeed, to the extent that consumer software is a consumer product under the Magnuson-Moss Act, the vendor is *already* required to allow customers to see the contract. *What's the big deal?*

The standby committee also explained that “commentators schooled in law and economics theory”¹⁴ argue that it is more efficient to provide terms with the product rather than allowing customers see them beforehand. For vendors, this probably is more efficient, but not for customers who want to compare terms in the course of their buying decision.

Withholding the terms until the customer is first trying to use the product, probably minimizes the probability that the customer will take the time to study the terms, consider their impact, compare terms across products, and then make a purchase decision that is informed by the comparison. This process also reduces the likelihood of detailed journalists' comparisons of licenses, because it makes the process of getting the licenses harder and more expensive. UCITA's definition of contractual use term is also relevant here. I provide the full text below, but the operative words in this context are: A “contractual use term . . . limits the . . . disclosure of . . . a term that defines the scope of a license.”

This will make it easier for vendors to slip outrageous terms past their customers—efficient for such vendors, but not for the customers.

2. *Nondisclosure of Known Defects*

In a marketplace in which warranties are routinely, non-negotiably disclaimed and remedies are reduced to virtually nothing, customers have a particularly strong interest in knowing what's wrong with the product before they buy it. Starting with journalist Ed Foster's early meetings with Ray Nimmer, the problem of nondisclosed known defects has been one of the top items on the mass-market customer's agenda.

This has also been a key issue for the software developers' professional societies.

The Institute for Electrical and Electronics Engineers (IEEE) wrote:

“UCITA allows software publishers to disclaim warranties and consequential damages even for software defects known to the publisher prior to sale, undisclosed to the buyer, and having damages that can be reasonably foreseen.”¹⁵

For example, the President of the Association for Computing Machinery (ACM), wrote to NCCUSL that:

“UCITA makes it too easy for software publishers to avoid facing any legal consequences for defective software. Perhaps this is appropriate for some defects, but not for the ones the publisher knew about when it sold the product. Customers can't discover most of these defects with quick trials of the program - it takes skill to find them during pre-use testing. By reducing the responsibility of software publishers to detect and eliminate problems before the product is released to the public, UCITA will result in the lowering of standards in our profession.”¹⁶

Within the computing profession, IEEE and ACM jointly have the stature of the American Bar Association. These are serious concerns, that should be taken seriously.

Several proposals were made to the UCITA drafting committee, ranging from detailed analysis of the nature of a material software defect¹⁷ to proposals that focused on damages (none for disclosed defects, none for defects not known to the vendor, and nondisclaimable but possibly capped for nondisclosed known defects).¹⁸

The UCITA standby committee rejected the idea that UCITA should require, or create incentives to encourage, vendors to reveal their known defects. Instead, they said, UCITA should defer to other law. Ultimately, UCITA Section 116 was amended to include the following statement:

“Among the laws supplementing and not displaced by this [Act] are trade secret laws, unfair competition laws, and the law of fraud, misrepresentation, and unfair and deceptive practices, including application of such laws as they may deal with failure to disclose defects.”

Unfortunately, it's not clear how much benefit, with respect to known defects, software customers gain from these laws.

Certainly, there is a firm requirement in many states that real estate sellers reveal known defects in the property that are not likely to be visible under a normal inspection. There have also been many automotive cases involving nondisclosed known defects.¹⁹

There have been very few successful cases involving nondisclosed known software defects. The most famous of them was the Intel Pentium lawsuit²⁰ which involved mathematical calculation errors on the Pentium chip, known but not disclosed by Intel until the defect was independently discovered and publicized. That case settled without yielding a published opinion. I'm not aware of any published cases on point.

The key problem with relying on a fraud theory is that the standard of proof is so high. In many cases there is no affirmative, false statement made to induce a buyer to enter into the transaction. There might not even be a directly relevant claim in the user documentation. For example, suppose that pressing a certain series of keys causes the program to halt in a way that corrupts data in the program's data file. There is probably no statement in that manual that this series of keystrokes is safe, or that the program will never corrupt its own data. Instead, there is merely a reasonable user expectation that the product will not corrupt its own data. Under these circumstances, I don't think there is a case to be made for fraud, even if the software publisher knew of the defect at the time of sale.²¹

There is more hope for the broader deceptive practices claims, but if it is true that a publisher is liable for deceptive trade practices if it releases products with significant, known, non-disclosed defects then every mass-market software publisher should have been in litigation (and lost) many times over. Instead, we have one famous case (Intel), nearly a decade old.

Rather than relying on a cause of action that may or may not be viable, advocates for customers and the engineering community urged UCITA to take up the issue, essentially creating a

nondisclaimable implied warranty that there are no nondisclosed, significant known defects. This was intended to counterbalance the UCITA-specific contract formation and terms disclosure rules that, for all practical purposes, eliminate warranties and remedies for defects in mass market products and allow vendors to charge customers for support in dealing with the defects, rather than reimbursing them for these types of expenses.

3. Applicability of Consumer Protection Rules

Several statutes apply to transactions in goods or in consumer goods. The best known of these is the Magnuson-Moss Act, which sets (for consumer goods) federal standards for disclosure of warranty terms, requires vendors to make the terms available to potential customers on request and prohibits disclaimer of implied warranties whenever there is *any* written warranty.²²

The Magnuson-Moss Act applies to all consumer goods. Consumer goods are those which are “normally used for personal, family, or household purposes.”²³ This is a broad definition.

According to the Federal Trade Commission:²⁴

"The Act applies to written warranties on tangible personal property which is normally used for personal, family, or household purposes. This definition includes property which is intended to be attached to or installed in any real property without regard to whether it is so attached or installed. This means that a product is a 'consumer product' if the use of that type of product is not uncommon. The percentage of sales or the use to which a product is put by any individual buyer is not determinative. For example, products such as automobiles and typewriters which are used for both personal and commercial purposes come within the definition of consumer product. Where it is unclear whether a particular product is covered under the definition of consumer product, any ambiguity will be resolved in favor of coverage."

No published court rulings have settled the question of the applicability of the Magnuson-Moss Act to software, but it is widely believed that courts would rule that the Act applies to consumer software.²⁵ The Software Publishers Association's *Guide to Contracts*²⁶ considered the applicability of the Magnuson-Moss Act and concluded that “It is reasonable to assume that software purchased for home computer use would be covered by the Act.” Similarly, the senior author of UCITA, Ray Nimmer, concluded that

“It is apparent that the provisions of Magnuson-Moss will affect a large number of software transactions in the mass market. The most obvious are the variety of educational and game software that are a major element of the microcomputer industry. Is it appropriate to lump all microcomputer software into a single category or is it permissible to distinguish among applications? It is arguable that only broad distinctions should be made. Most microcomputer or mass-market software is covered by the Act.”²⁷

In contrast, Fred Miller (then Executive Director of NCCUSL) claims that

“the Magnuson-Moss Warranty Improvement Act never was contemplated to apply to software as goods, and so to do that . . . seems to make no more sense than does leaving software transactions to be governed by Articles 2 or 2A or the common law if the form of the transaction does not involve goods.”²⁸

This contrast reflects a fundamental and unresolved ambiguity in UCITA. In section 104, UCITA appears to subordinate itself to all state and federal consumer protection laws. However, many of those laws state their scope as involving transactions in goods. If we define (as UCITA does) the mass-market software transaction as a license rather than a sale of a copy, then any laws that apply specifically to goods do not include the software transactions in their scope.

To clear up the ambiguity, the ABA working group recommended the language now found in UCITA Section 104 (f).

“104 (f) [**Determination of applicability of consumer protection law.**] The applicability of a consumer protection law is determined by that law as it would have applied in the absence of this [Act].”

Unfortunately, this still leaves the question open. If the transaction is a license, then the consumer protection laws that apply only to goods will not apply to the transaction, whether UCITA is in force or not.

One way to settle this would be consistent with the general approach of American courts for the last generation.²⁹ When considering cases involving off-the-shelf software, especially (according to Nimmer³⁰) in cases involving warranty issues, the courts have generally treated the transactions as if they were sales of goods and applied UCC Article 2. Within that framework, we would expect goods-related consumer protection laws to apply to mass-market software. And so, to clear up the ambiguity, we should amend UCITA to say that, for the purpose of determining the applicability of consumer protection statutes, a mass-market license of a software product should be interpreted as equivalent to a sale of a copy.

Alternatively, if the assumption is that traditional consumer protection laws should not necessarily apply to mass-market software, the better approach would be to have each state legislature walk through every consumer protection law and decide, statute by statute, to what extent each should apply to software, amending each one as appropriate. This is the approach taken by UCITA:

“104 (g) [**Applicable consumer protection statutes.**] Among the consumer protection laws of this State which apply to the subject matter of this [Act] are: [Insert statutes that, on review by the legislature and amendment as appropriate, are determined to be applicable to the subject matter of this [Act] such as a state's unfair and deceptive practices act with amendments as appropriate.]”

Across states, this seems likely to require an enormous amount of negotiation and unlikely to lead to any semblance of uniformity. Despite UCITA's apparent deference to the consumer protection laws, until those negotiations have taken place, we will have no idea to what extent traditional consumer protections will apply to mass-market software.

4. Scope of UCITA

The scope provisions of UCITA have been the subject of intense discussion and disagreement, not only on the policy but also on what the words actually mean.

The ABA working group recommended a simplification, adopted by NCCUSL, to drop what was a very complex opt-in provision, Section 104. Parties can still contract to bring a transaction within the scope of the agreement by applying Section 115's provision for varying the contract by agreement.

Despite this improvement, the scope of UCITA is still unclear and subject to strongly differing interpretations. Here are the key sections:

“102(a) (9) "Computer" means an electronic device that accepts information in digital or similar form and manipulates it for a result based on a sequence of instructions.

“102(a)(10) "Computer information" means information in electronic form which is obtained from or through the use of a computer or which is in a form capable of being processed by a computer. The term includes a copy of the information and any documentation or packaging associated with the copy.

“102(a)(11) "Computer information transaction" means an agreement or the performance of it to create, modify, transfer, or license computer information or informational rights in computer information. The term includes a support contract under Section 612. The term does not include a transaction merely because the parties' agreement provides that their communications about the transaction will be in the form of computer information.

“103 (a) [**Scope in general.**] This [Act] applies to computer information transactions.

“103(b) [**Mixed transactions.**] Except for subject matter excluded in subsection (d), if a computer information transaction includes subject matter other than computer information or subject matter excluded under subsection (d), the following rules apply:

“103(b)(1) [**Computer information and goods.**] If a transaction includes computer information and goods, this [Act] applies to the part of the transaction involving computer information, informational rights in it, and creation or modification of it. However, if a copy of a computer program is contained in and sold or leased as part of goods, this [Act] applies to the copy and the computer program only if:

“103(b)(1)(A) the goods are a computer or computer peripheral; or

“103(b)(1)(B) giving the buyer or lessee of the goods access to or use of the program is ordinarily a material purpose of transactions in goods of the type sold or leased.

“103(b)(3) [**All other cases.**] In all other cases, this [Act] applies to the entire transaction if the computer information and informational rights, or access to them, is the primary subject matter, but otherwise applies only to the part of the transaction involving computer information, informational rights in it, and creation or modification of it.”

The definition of a “computer” is very broad. Quickly surveying consumer electronics devices in my home, here are a few items that meet the definition of computer: a blood glucose monitoring system (blood sugar test device), a digital blood pressure monitor, a heart rate monitor (strap it around your chest to check heart rate while exercising). Do we *really* want UCITA’s easygoing rules to apply to the software that controls medical devices?

In an extended analysis of the scope of UCITA, Professor Phil Koopman and I listed several other examples of household devices that could be classed as computers or computer peripherals under UCITA.³¹

Another ambiguity associated with scope relates to its treatment of embedded software. Software is “embedded”, within the meaning of this statute, if it “is contained in and sold or leased as part of goods.” A manufacturer of goods who wants the embedded software to be covered under UCITA can accomplish this relatively easily by separating the software from the goods. For example, you can buy aftermarket fuel injector computers and software for your car. Splitting out other components is not technologically difficult. It is primarily a marketing / packaging decision.

The reason this ambiguity is important is that the contracting standards are so different between Article 2 and UCITA. For example, the doctrine of alienation still applies to goods. A seller can’t stop you from reselling your product to someone else. In contrast, UCITA specifically allows a no-transfer clause for software. If you make cars, and want to control the used market for your cars, you have an incentive to separately license the chips that control the car’s fuel injectors, suspension, brakes, and so on. The software that controls the car accounts for a substantial part of the cost and value of the modern car. Restrict the transfer of the car’s software and you make it impractical for the customer to resell the car.

The UCITA standby committee declined to change the basic scope formulation, saying that “the present scope has been directly or indirectly endorsed by numerous groups, including many who are not in the software industry, such as the National Electronic Manufacturers’ Association, the National Association of Manufacturers, and the American Electronics Association.”³²

Of course these groups would endorse this scope! It gives them a map for bringing the software embedded in their products under UCITA, rather than leaving the software under Article 2 and the goods-related consumer protection laws.

5. Public Comment

Another magnet for criticism is UCITA's definition of use restrictions:

"102(a) (19) "Contractual use term" means an enforceable term that defines or limits the use, disclosure of, or access to licensed information or informational rights, including a term that defines the scope of a license."

An example of a limitation on disclosure of licensed information is from Microsoft:

"You may not disclose the results of any benchmark test of the .NET framework component of the OS Components to any third party without Microsoft's prior written approval."³³

A similar clause, from Network Associates (McAfee Viruscan) is the subject of an unfair trade practices lawsuit, *New York v. Network Associates*.

NCCUSL adopted the following amendment:

"105(c) [**Lawful public comment not prohibited.**] In a transaction in which a copy of computer information in its final form is made generally available, a term of a contract is unenforceable to the extent that the term prohibits an end-user licensee from engaging in otherwise lawful public discussion relating to the computer information. However, this subsection does not preclude enforcement of a term that establishes or enforces rights under trade secret, trademark, defamation, commercial disparagement, or other laws. This subsection does not alter the applicability of subsection (b) to any term not rendered unenforceable under this subsection."

This provides less than might first appear.

- It applies only to software made available "in its final form." Does that exclude products that are frequently updated (think of all the security patches to Microsoft's Windows) and so, arguably, never in their final form? According to the Comments, no. However, the Comments are not part of the statute. They are not voted on by the legislatures. And unlike the process in, for example, the Article 2 drafting meetings, the UCITA comments were not carefully reviewed and publicly discussed as part of the drafting process.
- It protects speech only by "end-user licensees." In the November, 2001 standing committee meeting (which I attended), vendors complained of unfair reports by competitors and journalists. Are these the licensees who are excluded by this restriction? If so, are we not excluding the potentially most knowledgeable critics?
- Finally, there is the question of trade secrets. Many software licenses recite that the software contains valuable trade secrets. Every licensee clicks through the license agreement. If a vendor required every *user* to click through a license, and stated in the license that the operational details of the product were trade secret and that the user agrees to exercise best efforts to protect the secrets, what more would the vendor need to do in order to be able to bar negative reviews because they would reveal the company's trade secrets?

6. Reverse Engineering

Another common clause in software licenses, apparently enforceable as a contractual use term, is the ban on reverse engineering. This has drawn criticism from the professional societies and from individual developers and some corporate development groups.

NCCUSL adopted a new provision, Section 118, to deal with this:

“118 (b) [**Contractual term unenforceable.**] Notwithstanding the terms of a contract subject to this [Act], a licensee that lawfully obtained the right to use a copy of a computer program may identify, analyze, and use those elements of the program necessary to achieve interoperability of an independently created computer program with other programs, including adapting or modifying the licensee's computer program, if:

“118(b)(1) the elements have not previously been readily available to the licensee;

“118(b)(2) the identification, analysis, or use is performed solely for the purpose of enabling such interoperability; and

“118(b)(3) the identification, analysis, or use is not prohibited by law other than this [Act].”

Unfortunately, this section is very incomplete. Reverse engineering is done for many legitimate reasons beyond enabling interoperation between programs.³⁴ For example, one might reverse engineer a product to discover security vulnerabilities (the better to defend against them), to discover or repair defects.

7. Open Source and Free Software

Free software is generally described as software that is distributed free of traditional intellectual property restrictions. The Free Software Foundation identifies four key freedoms for users of free software:³⁵

- Freedom to run the program, for any purpose.
- Freedom to study how the program works, and adapt or improve it. (Inherent in this is making the source code available to the user.)
- Freedom to release improvements to the public.
- Freedom to redistribute copies. However, the new copies must be released under the same terms as the original.

Free software might be given away on its own or it might be distributed in conjunction with consulting services. Some free software is very complex and the average customer will need a consultant to help install it, train staff in its use, and perhaps modify it to meet the customer's needs. For an example, think about the Apache web server software, which runs more of the web than any other brand of server software. It is free, but a review of materials at www.apache.org might convince you that you might need some help setting it up at your law firm. What makes the program “free” is that the customer can use, examine, modify, and distribute the modified software without permission from or additional fee to the consultant or anyone else.

Similarly, to be “open source” software, the license must provide the user with access to the source code, a right to modify the program and distribute the original or modified version, and several other rights.³⁶

Under these definitions, commercial products that are given away for free but with no access to the source code, and no right to reverse engineer, modify or redistribute the software are neither free software nor open source. Thus, for example, Internet Explorer is not free / open software.

UCITA creates a new definition for “free software”:

410(a) [**Free software defined.**] In this section, "free software" means a computer program with respect to which the licensor does not intend to make a profit from the distribution of the copy of the program and does not act generally for commercial gain derived from controlling use of the program or making, modifying, or redistributing copies of the program.

UCITA does not attach the main implied warranties to software that it classifies as “free.”

Under NCCUSL’s definition, Microsoft Internet Explorer might well be designated as free software, but Apache will often not be free.

UCITA was written with minimal input from the free software community and extensive input from the commercial software community, especially Microsoft, which has been running a long public relations campaign disparaging and discouraging the use of free software.³⁷

A discussion of the needs of the free software community would require a separate article or series--it is beyond the scope of this one.

One of the key problems to consider, however, is that any free / open software that is circulating today will continue to be distributed after UCITA is passed, but the consultants who distribute the software cannot modify the license that shipped with the original copy of the software. The current licenses do not usually use click-to-assent or collect a formal manifestation of assent from the customer. They might well be unenforceable under UCITA.

Under the commercial software distribution model, the publisher can modify its license at will and stop selling copies under the old license. For a company like Microsoft, it is easy to disclaim warranties in a manner that will be enforceable under UCITA.

For a distributor of free software, the comparable decision is to stop distributing the free software products until the formalities of their licenses are adjusted by the original authors. Many of those will not even be available to do this--it is not unusual for someone to write a useful utility, release it as free software, and then let the community take over subsequent modification. The expectation is that a license that keeps the product free and recites the usual disclaimers will be sufficient.

The distributors of free software are often not wealthy people, nor legally sophisticated. For example, several computer science students do this kind of work (install and support free software) for \$10 or \$20 per hour. Much of the free software available is scientific or educational, not commercial. Unless we are very careful in drafting the rules that will govern free software licenses, we might find that impose unexpected business risks on public institutions, especially universities that post free software on their servers, educators, and very small businesses. The result would be to kill distribution of a massive number of products³⁸ that might never be replaced or might take years to replace. It is not clear to me that this would be a benefit to the public.

8. Clarity of UCITA

The UCITA drafting committee meetings, and the published debate surrounding them, were often acrimonious. People were accused of being liars. People ridiculed each others’ interpretations of UCITA.

The ABA Working Group flagged the underlying problem directly, saying that their “principal concern with UCITA, as presently drafted, is that it is extremely difficult to understand.”³⁹ They identified conflicts between the comments and the black letter, ambiguity in definitions, and concluded that “many of the ‘black letter’ rules come across as convoluted and, at times, inscrutable. Time and again, when the Working Group attempted to consider the substantive merits of a UCITA concept or provision, the Group had first to parse through the language word

by word and clause by clause, only to realize, in the end, that the individual members of the Group could not agree on what the particular section said or meant.”

The amendments have done little to clarify or disambiguate UCITA.

¹ Cem Kaner is Professor of Computer Sciences at The Florida Institute of Technology in Melbourne, Florida. Dr. Kaner is a member of the AFFECT coalition. However, this article is not intended as a presentation of AFFECT’s analysis of the amendments and Cem Kaner is not writing as a representative of AFFECT.

² AFFECT Coalition, Suggested Amendment to UCITA, http://ucita.com/pdf/ucita_amendments.pdf.

³ REPORT of UCITA Standby Committee, December 17, 2001, <http://www.nccusl.org/nccusl/UCITA-2001-comm-fin.htm>, hereinafter *UCITA Standby Committee 2001 Report*.

⁴ Vittone, John, M. Majority Report, ABA Member Working Group REPORT on the UCITA, January 31, 2002, http://www.abanet.org/ucita/report_on_ucita.pdf, hereinafter *ABA Working Group Report*.

⁵ *ABA Working Group Report*, “The following discussion assumes that all of the changes to UCITA recommended by the Standing Committee in December will be adopted. If that assumption proves to be incorrect, the Working Group would have still other areas of concern.”, page 6.

⁶ UCITA Standby Committee, Report to Conference and Response to Concerns Expressed in ABA Working Group Report, May 29, 2002, <http://www.nccusl.org/nccusl/ucita/Standby-cm-aba-resp.pdf>, hereinafter *UCITA Standby Committee 2002 Report*.

⁷ Amendments to Uniform Computer Information Transactions Act, <http://www.law.upenn.edu/bll/ulc/ucita/2002act.pdf>. The updated version of UCITA (2002 amendments included) is at <http://www.law.upenn.edu/bll/ulc/ucita/2002final.pdf>. Unless otherwise indicated, references to UCITA are to the 2002-amended version.

⁸ Henning, William H., Fred H. Miller, & Michael R. Kerr, UCITA Revisited, 48 UCC Bull. November 2002 at 1.

⁹ AFFECT Response to NCCUSL Commentary on UCITA 2002 Revisions, December 4, 2002, <http://ucita.com/pdf/AmendmentAnalysisFinal.pdf>.

¹⁰ This is not a controversial assertion, although it is surprising to many lawyers who have no software industry experience. I’ve made this assertion at dozens of seminars that I’ve presented to working software quality control staff, and it is routinely accepted and discussed -- with examples provided by the participants. For a specific example, “Microsoft Windows 3.1 was shipped in 1992 with some 5,000 known bugs” Edward Yourdon, *Rise & Resurrection of the American Programmer*, 1996 at 157. Another example: “Apple shipped the first version of Hypercard with about 500 known bugs in it, yet the product was a smashing success. The Hypercard QA team chose the *right* bugs to ship with. . . I was working at Apple when Macintosh System 7.0 shipped and it went out with thousands of bugs.” James Bach, *The Challenge of Good Enough Software*, (accessible after signing up for a free account at <http://www.veritest.com/tnetwork>). For further discussion on how to analyze problems as they are discovered in order to prioritize them and reduce the probability of releasing a product with unacceptable defects, see Cem Kaner, Jack Falk & Hung Nguyen, *Testing Computer Software* at 13, 73, 95-96, 109-111, 113-114.

¹¹ Some license terms get much more extreme than this. For example, on downloading an update to Windows, I received the following license term: “Digital Rights Management (Security). You agree that in order to protect the integrity of content and software protected by digital rights management (“Secure Content”), Microsoft may provide security related updates to the OS Components that will be automatically downloaded onto your computer. These security related updates may disable your ability to copy and/or play Secure Content and use other software on your computer. If we provide such a security update, we will use reasonable efforts to post notices on a web site explaining the update.” Another example comes from the Intel Photo Album II Applet, which I downloaded from Intel.com on May 23, 2000. (The applet is no longer available.) That one, a free download for consumers and businesses, took ownership of anything you created with the tool, “Licensee agrees that all works of authorship, inventions, improvements, developments making use of the Applet or any portion of the Applet, solely or in collaboration with others,

as well as all patents, copyrights, trade secrets, trademarks and other intellectual property rights therein and thereto (collectively, 'Developments'), are the sole property of Intel. Licensee agrees to assign (or cause to be assigned) and does hereby assign fully to Intel all such Developments." The cleverest new license terms grants a virus the right to propagate to everyone on your email list. Robert Lemos, Greeting Card Virus Licensed to Spread, November 13, 2002, <http://news.com.com/2102-1001-965570.html>.

¹² *ABA Working Group Report* at 18.

¹³ *UCITA Standby Committee 2002 Report* at 12.

¹⁴ *UCITA Standby Committee 2002 Report* at Footnote 4.

¹⁵ IEEE-USA Board of Directors, Position: Opposing Adoption of the Uniform Computer Information Transactions Act by the States, February 2000, <http://www.ieeeusa.org/forum/POSITIONS/ucita.html>.

¹⁶ Barbara Simons, Letter to NCCUSL, July 12, 1999, <http://www.acm.org/usacm/IP/usacm-ucita.html>.

¹⁷ Cem Kaner, What is a Serious Bug? Defining a "Material Breach" of a Software License Agreement, December 10, 1996, <http://www.badsoftware.com/uccdfect.htm>

¹⁸ Proposals along these lines came from the Institute for Electrical and Electronics Engineers, the Independent Computer Consultants Association, Ralph Nader's Consumer Project on Technology, and other participants in the drafting meetings.

¹⁹ See generally, Jonathan Sheldon, Carolyn Carter & Stephen Gardner, *Unfair and Deceptive Acts and Practices*, Fifth Edition (supplemented 2002), Sections 4.9.3 and 5.5.5.4, and Jonathan Sheldon & Carolyn Carter, *Consumer Warranty Law*, Second Edition (supplemented 2002), Sections 5.13, 11.3, and 14.4.3 (failure to disclose known material defects violate the vendor's duty of good faith).

²⁰ *In re Pentium Processor Litigation*, Master Case Number CV745729, 11 volumes on file at the Santa Clara County Superior Court, San Jose, California. See also W. Mossberg, Intel isn't serving millions who bought its Pentium campaign, *Wall St. J.* at B1, December 15 1994; Testing reveals that Intel's and IBM's claims both exaggerate the magnitude of problems, risks, 11 *PC Week* #50 at 1, December 19, 1994.

²¹ Under circumstances not much different from this, Timberline Software Corporation released a product with a known data-corruption defect that cost its customer \$1.95 million. *M.A. Mortenson Co. Inc. v. Timberline Software Corp.* 998 P.2d 305 (Wash. 2000). The case was dismissed on a summary judgment motion.

²² 15 U.S.C. Section 2308(a).

²³ *Id.* Section 2301(1).

²⁴ 16 C.F.R. Section 700.1(a).

²⁵ See L.J. Kuttan, *Computer Software: Protection/Liability/Law/Forms* § 10.03[3] (Vols 1-4, Release 24, 1999). As Kuttan put it while discussing the applicability of the Magnuson-Moss Act, "[s]ome software must be a consumer product. Otherwise society is caught in the dichotomy that (1) the hardware is a computer product, (2) the software is not a consumer product, yet (3) neither product can be used without the other."

²⁶ Thomas J. Smedinghoff, *The Software Publishers Association Guide to Contracts and the Legal Protection of Software* (1993) at 88.

²⁷ Raymond T. Nimmer, *Law of Computer Technology* (supplemented September 2002) at Section 6:69.

²⁸ Fred Miller, *Speaking Frankly About UCC Article 2B*, 37 *UCC Bulletin* #2, March 1999 at 1.

²⁹ Typical cases include *Advent Sys. Ltd. V. Unisys Corp.*, 925 F.2d 670 (3rd Cir. 1991), *Burroughs Corp v. Century Steel Inc.*, 664 P.2d 354 (Nev. 1983).

³⁰ Raymond T. Nimmer, *Law of Computer Technology* at section 6:4.

³¹ Philip Koopman & Cem Kaner, *The Problem of Embedded Software in UCITA and Drafts of Revised Article 2. Part 1*: 43 *UCC Bull.* #1, 1 February 2001, <http://www.badsoftware.com/embedd1.htm>, *Part 2*: 43 *UCC Bull.* #2, 1, March 2001, <http://www.badsoftware.com/embedd2.pdf>.

³² *UCITA Standby Committee 2002 Report* at 8.

³³ This has appeared in several Microsoft licenses, for example the Microsoft Windows Update (Critical Upgrade Notification 4.0).

³⁴ Cem Kaner, Article 2B and Reverse Engineering, 36 UCC Bull. #2, 1, November 1998, <http://www.badsoftware.com/reversea.htm>

³⁵ See, for example, the Free Software Foundation's description at <http://www.gnu.org/philosophy/free-sw.html>.

³⁶ See the Open Software Initiative's definition page at <http://www.opensource.org/docs/definition.php>. For our purposes, we can ignore the differences between "open source" and "free" software.

³⁷ See, for example, Craig Mundle, The Commercial Software Model, May 3, 2001, <http://www.microsoft.com/presspass/exec/craig/05-03sharedsource.asp>. Some of Microsoft's tools now include use restrictions that forbid developers from using the tools to create free software. Stephen Shankland, Microsoft License Spurns Open Source, CNET news.com, June 22, 2001, <http://news.com.com/2100-1001-268889.html>.

³⁸ Look for yourself at the types of products developed as free software. See, for example, www.sourceforge.org. Many of these would never be commercially viable..

³⁹ *ABA Working Group Report* at 7.