

WHY SOFTWARE QUALITY PROFESSIONALS SHOULD ACTIVELY OPPOSE THE UNIFORM COMPUTER INFORMATION TRANSACTIONS ACT

Copyright © 1999, Cem Kaner

Draft, May 15, 1999

The Uniform Computer Information Transactions Act (UCITA) is a proposed new law that will govern all transactions in software, including contracts for sale, licensing, documentation, maintenance and support of computer software. It will also govern contracts involving electronic information (movies, music, text that you download or buy on a CD) and, at the vendor's option, can govern sales of computers and some other devices that are sold in conjunction with software.

Until recently, UCITA was proposed as an amendment to the Uniform Commercial Code, and was called Article 2B. However, the American Law Institute (ALI), one of the two organizations that must approve all changes to the UCC, recently withdrew from the Article 2B project. The other organization, the National Conference of Commissioners on Uniform State Laws (NCCUSL), decided to rename the bill to the Uniform Computer Information Transactions Act and go forward with it.

SIDEBAR

The American Law Institute has declined to say why it withdrew from the Article 2B project. However, their withdrawal was not a surprise.

In its May 1998 Annual Meeting, the ALI passed the following resolution: "The current draft of proposed UCC Article 2B has not reached an acceptable balance in its provisions concerning assent to standard form records and should be returned to the Drafting Committee for fundamental revision of the several related sections governing assent."

The authors of the ALI resolution (Braucher and Linzer) wrote in their supporting memo that "The Draft reflects a persistent bias in favor of those who draft standard forms, most commonly licensors." (Companies that publish or sell software are licensors under 2B.)

Additionally, in December 1998, an ALI Council ad hoc committee formed to review Article 2B submitted a memorandum to the full Council stating that it was unlikely that an acceptable draft could be prepared in time for the ALI Annual Meeting in May 1999. The memorandum raised questions about whether the project is premature in light of rapidly changing technology and business practices. It also noted the lack of consensus about the need for Article 2B and the opposition to it from many affected interests. The memorandum described the drafting of key provisions as "opaque" and "difficult to comprehend."

For more details, see Braucher, 1999.

NCCUSL will vote on UCITA at its annual meeting, July 23-30, 1999 in Denver. There will be a final UCITA drafting committee meeting on July 22, 1999 in Denver. For details on the meeting, contact me at kaner@kaner.com or check www.nccusl.org.

If you read a copy of this article before July 22, 1999, I urge you to write a letter to the members of NCCUSL from your State, asking them to oppose UCITA. If you read it in the July 20-30 timeframe, you can send a letter to me and I will see that it reaches the NCCUSL members from your state. After July 30, send opposition letters to your state representatives.

Why We Should Actively Oppose UCITA

The simple and short answer is that UCITA will dramatically reduce a software publisher's external failure costs for defective software. It does this brilliantly, in a wide range of ways, reducing the costs of customer support, of lost sales due to competition, and of legal action.

As a result, UCITA changes the economics of software publishing.

When we reduce the risks (to the publisher) of selling defective software, we reduce the incentive to spend the money and time to prevent, search for, and fix defects. In turn, this tells me that we (the American software industry):

- will ship worse software.
- will invest less money in technology and process improvement needed to produce better software.
- will make the American industry more vulnerable to foreign competition. Les Hatton, a well known author on software quality (see, for example *Safer C*), just finished his masters degree in law. He advises me (personal communication, May 13, 1999) that the trend in Europe is to hold software companies more accountable for defects and to provide greater protection for European consumers and small businesses. I believe that this will provide a greater incentive for companies that primarily trade in Europe to improve their products, rather than ship them with obvious defects. Eventually, international competition will take care of this divergence of standards. But as with the car industry, that eventuality might be devastating for some parts of the American economy (us — software workers — for example).

What We Can Do

For the last three and a half years, I've spent about one-third of my time, unpaid, explaining software quality issues to legislative drafting and regulatory bodies. I've provided input to drafting committees for Uniform Laws (Article 2B/UCITA, Article 2—UCC law of sales, and the Uniform Electronic Transactions Act), to people drafting laws and treaties to govern international electronic commerce (a State Department study group, and members of the American Bar Association), to the Federal Trade Commission, and to various consumer protection groups. Several other software quality advocates have shared in this work, including Watts Humphrey, James Bach, Doug Hoffman, Sharon Marsh Roberts, Melora Svoboda, Ken Pugh, Brian Lawrence, and Bob Johnson. Software-related professional societies, including the Association for Computing Machinery, the Institute for Electrical and Electronics Engineers, the Independent Computer Consultants Association, and the software-test-discuss mailing list (but not ASQ) have submitted letters criticizing UCITA/Article 2B.

Here are a few things that I've learned.

First, UCITA is just one of several legislative proposals involving software quality that will go to the state and federal governments over the next few years. It is currently the most important. I expect to also see proposals to:

- reduce legal liability of vendors and users of Y2K-defective software;
- license software testers, developers, and consultants;

- increase liability for defects in consumer or mass-market software (various proposed lemon laws);
- limit competition, via changes to the Copyright Act to (for example) ban or restrict reverse engineering of software products;
- increase competition, via changes to the antitrust laws (if Microsoft prevails in its antitrust case);
- increase / decrease / regulate / deregulate privacy protection on the Net;
- establish standards for reliability of internet services;
- establish standards for electronic signatures and other technical aspects of electronic commerce.

There will probably be plenty of other proposals.

Second, we are credible sources of information on these types of issues. We are industry insiders. We aren't embittered whistleblowers—we want the industry to succeed. We also have special insight—we know how products fail, we understand the difficulties of making perfect products and we also know how our defects affect customers.

Our input is valuable because most of the people who will have to evaluate these proposed laws are lawyers, and most of them are unsophisticated about software. Many of the lawyers working on committees writing legislation to govern electronic commerce didn't even have e-mail accounts when they started work. Many of the lawyers who will vote on these proposed laws still don't have e-mail, let alone more sophisticated e-commerce experience.

Even the ones with some experience as software users get a huge proportion of their education about software development and marketing from other lawyers who represent software publishers. This bias is pervasive. Legislative drafting committees dealing with software are visited or advised by many paid lobbyists for software publishers and by very few, usually unpaid, consumer advocates (almost none of whom have software-related backgrounds). Additionally, courses and industry seminars on software law are typically taught by lawyers who represent software publishers / consultants. And speakers at conferences on software law are typically lawyers who represent publishers. There are hundreds of lawyers working for software sellers, but I can count the number of lawyers who publicly advocate for software quality on one hand.

If legal drafting bodies and legislatures are going to deal sensibly with the proposed laws to govern software quality, they need input from people like us.

Third, we can provide input—we are *welcome* to provide input—as individuals and as professional societies. People are hungry for our input. Non-lawyers can have a significant impact on laws by addressing technological issues and explaining the consequences of technology-related decisions. Software developers and testers haven't been all that well received in the UCITA/Article 2B drafting committee meetings, but they have had big effects elsewhere. For example, Bob Johnson is responsible for many significant improvements to the Uniform Electronic Transactions Act. And, even in the UCITA process, our comments have been effective in slowing the process down and convincing decision-makers to consider UCITA more carefully. ALI would probably have approved 2B/UCITA if it wasn't for our many comments.

In my experience, regulatory agencies, such as the Federal Trade Commission, are even more interested in our input than legislative drafting groups.

With particular reference to UCITA, we can do the following:

- Write letters to the head of NCCUSL, Gene Lebrun, <GLEbrun@LYNNJACKSON.COM>. (Please send me a copy so that I can distribute them to the rest of NCCUSL at the annual meeting in July, 1999. It is unlikely that a letter to Gene will go further, and he strongly supports UCITA.)
- Write letters to the NCCUSL members in your state (contact me, kaner@kaner.com, for their names, etc. or check my website, www.badsoftware.com).
- Write letters to your state legislators and state governor. (This is a state law issue; members of Congress don't count.)

- Write letters describing defects that were badly handled and examples of deceptive or dishonest or unfair conduct by software publishers to Adam Cohn <ACOHN@FTC.GOV> of the Federal Trade Commission. Adam will of course be interested in fully detailed (names, dates, etc.) letters. You can also send him a letter that deliberately disguises the people/companies/products, that is sent to him only to educate him about the types of practices in the industry. Tell him, in these letters, that this is what you are doing and (if you want) tell him that I suggested that he would find these educational-purposes-only letters useful. These are valuable because the FTC is in an awkward position regarding UCITA. Federal agencies rarely comment on state law, but the authors of UCITA are making claims about the status and content of consumer protection law in the USA, and the FTC has significant expertise in this area. The FTC wrote one long letter commenting on Article 2B (see www.ftc.gov/be/v980032.htm) but has to decide whether to write another and which issues are important to address.
- Write letters and op-ed articles for your local newspaper. I can help you a bit with this.
- Encourage your professional societies (such as ASQ) to take a stand and to write some letters of their own.

=====

SOME ORGANIZATIONS THAT OPPOSE UCITA

- fifty intellectual property law professors (www.2BGuide.com/docs/1198ml.html)
- American Association of Law Libraries (www.arl.org/info/letters/libltr.html and www.arl.org/info/letters/Wright_ALI_letter.html)
- American Library Association (www.arl.org/info/letters/libltr.html and www.arl.org/info/letters/Wright_ALI_letter.html)
- American Society of Media Photographers (www.nwu.org/pic/uccasmp.htm)
- Association for Computing Machinery (www.acm.org/usacm/copyright/usacm-ucc2b-1098.html)
- Association of Research Libraries (www.arl.org/info/letters/libltr.html and www.arl.org/info/letters/Wright_ALI_letter.html)
- Consumer Federation of America (www.cptech.org/ucc/sign-on.html)
- Consumer Project on Technology (Ralph Nader) (www.cptech.org/ucc/sign-on.html)
- Consumers Union (www.2BGuide.com/docs/cu1098.html)
- Independent Computer Consultants Association (unpublished)
- Institute for Electrical & Electronics Engineers (IEEE) submitted specific criticisms of 2B but not final opposition. See www.ieee.org/usab/FORUM/POLICY/98feb23.html and www.ieee.org/usab/FORUM/POLICY/98oct09.html.
- Magazine Publishers of America (www.2BGuide.com/docs/v9-98.pdf)
- Motion Picture Association of America (www.2BGuide.com/docs/v9-98.pdf and www.2BGuide.com/docs/mpaa1198.html)
- National Association of Broadcasters (www.2BGuide.com/docs/v9-98.pdf)
- National Cable Television Association (www.2BGuide.com/docs/v9-98.pdf)

- National Consumer League (www.cptech.org/ucc/sign-on.html)
- National Music Publishers Association (unpublished)
- National Writers Union (www.nwu.org/pic/ucc1009a.htm)
- Newspaper Association of America (www.2BGuide.com/docs/v9-98.pdf)
- Recording Industry Association of America
(www.2BGuide.com/docs/v9-98.pdf and
www.2BGuide.com/docs/riaa1098.html)
- Sacramento Area Quality Association (unpublished)
- Society for Information Management
(www.2BGuide.com/docs/simltr1098.html)
- software-test-discuss (this is the Net's largest e-mail discussion forum on software quality control)
- Special Libraries Association (www.arl.org/info/letters/libltr.html and
www.arl.org/info/letters/Wright_ALI_letter.html)
- United States Public Interest Research Group (www.cptech.org/ucc/sign-on.html).

Most of these letters are brief. After consultation with some other consumer advocates, I submitted a detailed letter with a section-by-section call for consumer-side revisions (www.badsoftware.com/kanerncc.htm). The Society for Information Management's letter details the concerns of large software customers (www.2BGuide.com/docs/simltr1098.html).

How UCITA Drives Down Failure Costs

The total quality cost for a product is the sum of:

- prevention costs (cost of preventing defects) plus
- appraisal costs (such as cost of testing)
- plus internal failure costs (such as cost of fixing defects)
- plus external failure costs (costs caused by the defect after the product is released to customers).

The external failure costs in this model are the costs of the seller or manufacturer, not the costs of the customer. This model ignores the customer's costs (Kaner, 1996).

Normally, the best way to reduce external failure costs is to improve the product, especially by preventing defects or by finding them early in development. However, a company *can* reduce its external failure costs by handling them (e.g. customer complaints or lawsuits) more efficiently.

UCITA provides another approach—reduce external failure costs directly.

I classify external failure costs into three categories:

- Customer support costs
- Legal costs.
- Lost sales (especially sales lost to competitors).

Note that the publisher doesn't reduce its customer's losses by reducing these costs. In many cases, the publisher will save money by *increasing* its customer's losses under UCITA.

Customer Support Costs

Here are some of the ways that UCITA lets publishers reduce their technical support costs (without improving the product). Citations are to the July, 1999 draft of UCITA:

- The publisher gets to charge customers for support, even for known bugs. For example, if you buy a program for \$50, the publisher might charge you \$3 per minute for a support call. Suppose that you run into a (known) defect, call the publisher, talk for 30 minutes (\$90), realize that you're not getting anywhere, and demand a refund. The publisher says OK, you send back the product (at your expense), the publisher sends you \$50 and keeps your \$90. It would have been much cheaper to throw the defective product away. (Section 803(a)(1) or 803(c).)
- When the buyer rejects a defective product because of obvious defects, the publisher can demand "a full and final statement in a record of all defects on which the refusing party proposes to rely." (Section 702(c)(2).) If there's a bug that you don't find and report in response to this, you can't complain about it when it bites you later. (But not even the publisher's testing staff can find all the defects, so why should we expect a customer to be able to create a full and final statement of them?)
- A publisher's contract to support its software will not require it to fix all defects. (Section 612(a)(1)(B).)
- In a contract dispute, the publisher can sometimes use "self-help" to shut down the operation of the program without court approval. (Section 816.)
- The publisher will have the same or (probably) greater power to restrict customer's right to maintain the publisher's software or to contract for 3rd party support for the software. . (This is achieved via contractual use restrictions on modification or 3rd party use of the product, see Section 102(a)(20) and 701(a).)

Legal Costs

Here are some of the ways that UCITA lets the publisher reduce its risk of legal liability for defective products, without making the product less defective.

- All of the terms of the contract except the price can be hidden from the customer until after the sale. By "hidden", I mean that the customer has no way to obtain the terms until after paying for the product. (Section 112, 210-213.)
- The implied warranty of merchantability (which provides that products will be reasonably fit for ordinary use) will be trivially easy to disclaim (to refuse to provide), and the disclaimer can be hidden from the customer until after the sale. (Sections 112, 210-212)
- By defining software transactions as licenses (which are intangibles) instead of sales of copies of the software (Section 102(a)(42)), UCITA takes these transactions out of the scope of the Magnuson-Moss Warranty Improvement Act and of state consumer protection statutes that are based on sales of goods go away. Consumers thereby lose warranty rights.
- It is much harder under UCITA than under current law (UCC Article 2) to prove that a warranty was created by a demonstration of a product. (Section 402(a)(3) and 402(b)(2)). A publisher can more easily get away with making misleading product demonstrations at trade shows.
- Business customers lose their right to reject a product for obvious defects. (Section 704(b) restricts the centuries old "perfect tender rule" to mass market contracts, repealing it for the rest.)
- Except for mass-market software in the first day or so of use, you cannot cancel the contract (and return the software) unless there is a "material breach" (a very serious defect or group of defects). (Section 601, 704(a), 802(a).) The definition of material breach (Section 701) is more seller-friendly than the current definition, as laid out in the *Restatement of Contracts*. And finally, the publisher can specify that you cannot cancel the contract even if there is a material breach. (Section 803(a)(1).)

- Even if it is proved that the product is defective and the seller has materially breached the contract, the seller is liable for almost no remedies (payments to the customer). For example, the publisher doesn't have to reimburse callers for incidental expenses (such as the cost of phone calls to the publisher or of returning the product) or consequential losses (such as the cost of restoring lost data) caused by the product's defect. (Section 803(d).) UCITA eliminates the principle of the "minimum adequate remedy" developed in UCC Article 2 (the current law of sales, which governs contracts for packaged software today -- See Reporter's Note 6 to Section 803: "This Act does not give a court the right to invalidate a remedy limitation because the court believes that the imitation does not afford a "minimum adequate remedy" for the aggrieved party.").
- The publisher can easily set up a waiver of liability (you "agree" to not sue the publisher for defects that you have complained about) by including the waiver in the click-wrapped "license" that comes with a bug-fix upgrade that the publisher sends you. (Section 702(a).)
- The contract can specify what state or country's law will govern this transaction and what court (in what country, state, city) you have to go to in order to bring a suit against the publisher. Forget about bringing a case in small claims court against a publisher who sells you a defective consumer product. Unless the software is very expensive, or the suit is brought as a class action, you probably won't be able to afford to bring such a lawsuit because of the added travel and legal research expenses. Additionally, the publisher will probably have written into the contract the state whose laws and courts are most favorable to it. (Sections 109, 110, and many debates and resolutions during the 2B drafting meetings.)

Lost Sales (Especially Sales Lost to Competitors)

- UCITA will let companies prohibit publication of criticisms of their product. For example, with VirusScan, you get the restrictions, "The customer shall not disclose the results of any benchmark test to any third party without McAfee's prior written approval" and "The customer will not publish reviews of the product without prior consent from McAfee." These are restrictions on disclosure. Section 102(b)(20) defines a "contractual use restriction" as "an enforceable restriction created by contract, which restriction concerns the use or disclosure of, or access to licensed information or informational rights, including a limitation on scope or manner of use." Therefore, on their face, these terms are enforceable. Section 105 provides some limitations on these clauses. A clause can be thrown out if federal law specifies that it can be thrown out or if the customer can jump through a remarkable set of litigation hoops to prove that the clause violates a fundamental public policy or if (Section 111) a judge rules that the clause is unconscionable. These decisions are made on a case by case basis, so it will probably take many years, many trials, and many appeals before we have a clear understanding of which restrictions are enforceable and which are not. In the meantime, publishers of defective products will be able to intimidate people who can't afford to be sue, by quoting their nondisclosure terms in their contracts and threatening to sue if the person publishes a critical article. (Such threats *have* been made.)
- UCITA will let publishers ban reverse engineering. This is just another contractual use restriction (102(a)(20)). See the discussion of these in the point above. There are many legitimate reasons for doing reverse engineering, such as figuring out how to make one product compatible with another, figuring out how to work around the defects in a product, and figuring out how to fix the product's defects. (See Kaner, 1998, for discussion and a list of other examples.)
- Finally, by making it easy for publishers to hide the terms of their contracts until after the sale is made, UCITA makes it hard for customers to compare several types of terms. For example, when you buy a program, do you know whether that publisher's warranty is better than its competitors'? What does the publisher charge for support, compared to its competitors? What are your rights to arrange for 3rd party maintenance of the product? Can you write critical reviews of it? These terms might all affect your buying decision, but not if you can't find them until after you've paid for the product.

In Closing

I've focused this paper on UCITA's impact on software quality, but UCITA has many other serious problems involving electronic commerce and intellectual property rights. If you want details, or if you can offer some help, please write me (kaner@kaner.com).

References

Braucher, J. (1999) "Why UCITA, Like UCC Article 2B, is Premature and Unsound", forthcoming in the *UCC Bulletin*, July 1999. Available at <http://www.2BGuide.com/docs/0499jb.html>.

Kaner, C. (1996), "Quality cost analysis: Benefits and risks", *Software QA*, Volume 3, #1, p. 23, www.kaner.com/qualcost.htm.

Kaner, C.(1998), "Article 2B and reverse engineering", *UCC Bulletin*, November, p. 1, www.badsoftware.com/reversea.htm. See also "The problem of reverse engineering", *Software QA*, www.badsoftware.com/reveng.htm.