Berkeley Technology Law Journal

Volume 13
Issue 3 *Fall*

Article 11

September 1998

On Self-Enforcing Contracts, the Right to Hack, and Willfully Ignorant Agents

James Raymond Davis

Recommended Citation

James Raymond Davis, *On Self-Enforcing Contracts, the Right to Hack, and Willfully Ignorant Agents*, 13 Berkeley Tech. L.J. 1145 (1998).

Link to publisher version (DOI)
https://doi.org/10.15779/Z38X68X

ON SELF-ENFORCING CONTRACTS, THE RIGHT TO HACK, AND WILLFULLY IGNORANT AGENTS

By James Raymond Davis †

ABSTRACT

Various aspects of the proposed draft of Article 2B have been critiqued from a number of different perspectives. This Article comments on Julie Cohen's description of self-enforcing contracts and the technical feasibility of Cohen's proposed "right to hack." Finally, certain provisions of Article 2B are analyzed from the standpoint of a software practitioner—a self-proclaimed hacker.

TABLE OF CONTENTS

I. INTRODUCTION

I must begin with a disclaimer. I am not a lawyer, economist, or policy analyst. I am a hacker, by which term I mean one skilled in the art of software, as opposed to a cracker, one who uses hackerly skill for illicit profit or wanton destruction. The distinction between the two will be important to part of the argument below.

In this brief note, I will discuss certain aspects of Article 2B from a technologist's point of view. First, I will extend the analogy that Julie Cohen offers for self-enforcing contracts,[1] showing that it raises even more problems than the one she suggests. Second, I will raise some doubts regarding the technical feasibility, and hence value, of the "right to hack"

---

† Jim Davis received his doctoral degree from the Media Lab at the Massachusetts Institute of Technology. He is a research scientist and hacker at Xerox's Palo Alto Research Center and was the original architect of the Dienst distributed library, which is running at roughly 100 institutions world wide. The comments expressed by Dr. Davis in this article are his own, and do not necessarily reflect the opinions of Xerox, Inc.

1. *See* Julie Cohen, *Copyright and the Jurisprudence of Self-Help*, 13 BERKELEY TECH. L.J. 1089 (1998).

that Professor Cohen proposes.[2] Finally, I will offer a criticism, from the standpoint of the software practitioner, of the section of Article 2B which deals with negotiating with software agents.

II. FURTHER CAPABILITIES OF SELF-ENFORCING CONTRACTS

Julie Cohen compares self-help repossession to a "repo man" who enters your house to repossess a sofa using a device like the "transporter" on the science-fiction television series *Star Trek*.[3] Consistent with Cohen's use of *Star Trek* imagery, I propose that "Odo" from the *Star Trek* sequel *Deep Space Nine* provides a more suitable analogy. This character, like Proteus of Greek myth, has the ability to assume any shape he wishes. A self-enforcing contract, then, can be compared to Odo, after he has assumed the shape of a sofa. Odo does not need to "beam" into your house to enforce the terms of the contract, because he was there from the moment you brought the sofa into your house. Admittedly, such an "intelligent sofa" is unlikely to reach the market soon. A more tangible example of a self-enforcing contract is a rental car with contractual terms that prohibit driving at speeds greater than 75 miles per hour, and whose electronic engines strictly enforce this limit. Here the actual contract and enforcement are separate, but in the digital products that will soon be commonplace, the two entities will be one. These examples of "embedded intelligence" suggest a number of additional consequences, which may arise much sooner than expected as computer intelligence becomes a more integral part of the goods and services industry.

An intelligent product, be it a sofa, a car or a word-processing tool, can do more than just assist in self-help repossession. First, intelligence allows a vendor to set and enforce arbitrary limits on capability. Imagine a sofa that automatically cancels the warranty if more than three adults sit on it. In the software industry, one common distribution tactic is to provide a commercial product, free of charge, that only contains a subset of the full range of features. The limitations of this type of software, known as "cripple-ware," are always fully disclosed, because the entire point of the strategy is to demonstrate that these limits could be removed for a fee. However, it is less clear that limits imposed by self-enforcing contracts would be as fair, reasonable, or obvious to the buyer.

Second, an intelligent product can try to understand what you do with it: how, when and why you use it. What happens next depends on the ca-

2. *See id.* at 1141.
3. *See id.*

pabilities of the agent. The product may simply offer advice, as does the "paper clip agent" in Microsoft Windows 97. It might suggest additional products for you to purchase. It might even contact a vendor on your behalf. This may be seen as helpful, or it may be taken as a violation of privacy. Suppose, for example, that while you are editing a letter about your plans for a forthcoming vacation, your intelligent word-processor relays a summary to a travel agent.

Finally, an intelligent product can change its price structure. An intelligent sofa might be sold at a low base price, then charge you only for the actual hours you spend on it. A very intelligent sofa might change its price depending on its model of your willingness and ability to pay. Got a hot date for Friday night? Better hope the sofa doesn't know, or the price might rise! I return to the topic of negotiating with software agents in the final section of this note.

One could argue that the capabilities I have outlined could lead to a more equitable distribution of goods and services. While I do not dispute this, it is important that these particular consequences be addressed in any discussion of Article 2B implications.

III. THE "RIGHT TO HACK" IS NOT LIKELY TO BE USEFUL

Professor Cohen argues for a "right to hack" as a means of achieving access to which one is legally entitled, and would otherwise have, except for software obstruction.[4] I sympathize with her, but technically, I am highly skeptical that this right will be of any use, even if granted. My argument stems from the relative advantage of those creating and using software security over those who would hack them. The owners of a valuable resource will expend as much effort to protect the resource as they would expect to lose were it left unprotected, less the cost of such protection. But the cost of this protection can be amortized over the full product line. By assumption, Cohen's hackers are only seeking use to which they are already entitled; they are not, for example, seeking to divert the entire revenue stream from the sofa, or to make a thousand copies. Rationally, these hackers should only exert as much effort as the missing value. But, attacks on modern security systems are expensive, and hence only rational when either the value is high (diverting electronic funds transfer) or where many potential customers are available to share the costs. Such conditions will rarely exist for hackers.

---

4. *Id.*

IV. ON WILLFULLY IGNORANT AGENTS

Article 2B section 204(3) states that when a human individual is negotiating with a software agent, the contract "terms do not include terms provided by the individual in a manner to which the agent could not react."[5] It is unclear to me, as a software practitioner, what is meant by "a manner in which the agent could not react." The abilities of a typical software agent to understand and react will be limited more by the effort expended by its creator than the state of the art. Article 2B, as written, places no obligation on the agent's maker to support reasonable forms of understanding, or even to provide an explicit indication of incomprehension. The agent can simply remain mute, and thereby nullify proposed terms. This defies the principles of user interface design which are known to every skilled practitioner.[6] A user interface should provide either a positive indication of comprehension or an explicit indication of non-comprehension. Although an explicit indication may be frustrating and uninformative about the source of the problem,[7] it leaves the user with no doubt as to whether the communication was accepted or rejected.

To a large extent, the lengths to which one must go in providing such indications depend on the user's familiarity with a particular device. I would expect most readers of this journal to have used voice interactive systems (e.g., voice mail, reservations, or home banking) with enough frequency as to be familiar, if not delighted, with their conventions. For instance, most of us know that it is useless to speak; that we must instead push buttons, and then only when prompted to do so. But this was not always common knowledge. In 1986, I worked on an early voice interaction system.[8] At that time, it took substantial effort to make this interaction work at all. It was difficult to engage people in a dialog, because unlike an answering machine, such systems speak several times, not just once. Further, it was hard to convince people to respond by pressing buttons, not by speaking, as they would to an answering machine. Were I to deploy such a system today, my task would be simpler, as I could rely on a base of shared cultural experience in using such systems. Still, if cost were no object, or if the interaction were sufficiently important, I would include cir-

---

5. U.C.C. § 2B-204(3) (Aug. 1, 1998 Draft).

6. *See generally* Philip J. Hayes & D. Raj Reddy, *Steps Toward Graceful Interaction in Spoken and Written Man-Machine Communication*, 19 INT. J. MAN-MACHINE STUDIES 231 (1983).

7. For example, the much maligned "syntax error" message.

8. James R. Davis, *Giving Directions: A Voice Interface to an Urban Navigation Program*, AMERICAN VOICE I/O SOCIETY PROC. (1986).

cuitry to detect when my users were trying to talk, so that an agent could advise them that their input was not understood.

In summary, there are two important issues to consider when agents are involved in negotiation. First, the standard of "could not react" is not only poorly defined in Article 2B,[9] but is also dependent on the amount of care the vendor chooses to take on the limits of technology. For instance, a vendor may cut corners by building a system that would not be able to react appropriately. Should this be a way to avoid negotiation? Second, the vendor's considerable freedom in setting the limits of an agent's ability to negotiate makes it unreasonable to place the burden of discovering and coping with such limits on the user. Instead, a "reasonable person" standard should be used, i.e., what a reasonable person knows or should know about an agent's capabilities. Since this standard will change as the population becomes more familiar with intelligent agents, vendors who wish to obtain certainty would do well to follow the principles established by the user interface design community. After all, if we are going to have Odo in the couch anyway, we should not have him "play dumb."

9. U.C.C. § 2B-204 Reporter's Note 1 (Aug. 1, 1998 Draft).