
Tod M. Turley
EXPERT SOFTWARE SYSTEMS: THE LEGAL IMPLICATIONS

I. INTRODUCTION

For those of us fascinated with HAL, the *2001: A Space Odyssey* computer\(^1\), the viability of such a machine is on the near horizon. Conceptually, a computer such as HAL can solve problems intelligently like a human expert and HAL also reasons using its deductive process based on a knowledge base\(^2\) that continuously expands by its own conclusions and by additional input from outside sources.

A. EXPERT SYSTEM DEFINED

The computer jargon often used to describe such a machine is an expert system.\(^3\) Expert systems are computer programs that have been constructed with the assistance of human experts which are capable of functioning at the standard of, or sometimes at a higher level than, experts in given fields. An expert system works by applying deductive principles to data contained in its knowledge base. The expert system embodies a depth and richness of knowledge that permit it to perform at a level equivalent to, or exceeding that of, an expert.\(^4\)

Expert systems are usually structured so that: (1) the user can request explanations of the lines of reasoning that lead to the expert system's conclusions, (transparency); (2) they reason with the informal, judgmental, experimental and potential knowledge that underlies expertise in a given field (heuristic); and (3) they allow, without great difficulty, modification to their knowledge bases (flexibility).\(^5\)

---

1. HAL refers to a spaceship computer in the movie *2001: A Space Odyssey*. HAL was able to build and add to its knowledge base (see infra note 2) and was capable of taking action independent of human programming and intervention. The name “HAL” was derived by replacing each letter of IBM with the letter immediately preceding it in the alphabet.

2. A knowledge base is a compilation of data stored in a medium that the computer can access at any given time.


4. For a more in depth discussion of what an expert system is, see Godall, supra note 3.

B. CURRENT DEVELOPMENT PROBLEMS

Most expert systems are still experimental. Software developers are increasingly investing in research and development of expert systems. Unfortunately, "[e]xpert systems software development costs are high, development times seem unusually long, and the resulting programs put a heavy burden on computing resources." The development problems are usually attributable to the lack of understanding of the human brain's reasoning process and to the inability to adapt that understanding to a machine. Because physical and social scientists do not definitively understand the human creative process, it is conceptually difficult for a computer programmer to create a representative computer language which emulates human reasoning. Also, along with the conceptual difficulties, effective expert systems require speed and storage capacities that modern machines cannot yet handle.

C. CURRENT EXPERT SYSTEM APPLICATIONS

Few expert systems are in day-to-day use. One of the most often mentioned programs is MYCIN. MYCIN provides assistance to physicians in diagnosing and treating bacterial infections. MYCIN is transparent because it lists the sequence of deduction that led to its conclusion. MYCIN and other medical diagnostic type expert systems have been very successful.

Inspired by the success of expert systems such as MYCIN, we can expect development in many areas such as psychology, law, engineering, architecture, business, aviation, and various others. Nevertheless, the uses of current expert systems are typically limited to specialized tasks.

---


7. The gains in computer processing speeds over the last five years have been phenomenal. Faster hardware will enable expert systems to handle larger knowledge bases and arrive at conclusions more quickly. The recent development of optical disk storage opens a new horizon for mass storage capacity for both mainframe and personal computers. Optical disks may adapt well to the storage requirements of the knowledge base in certain expert systems. Optical disks, however, are currently "read and write only" and are not yet ideally suited for expert systems storage requirements.

8. See Susskind, supra note 5, at 174. MYCIN was developed by the physician, Edward H. Shortliffe. The program provides consultive advice on diagnosis and antibiotic therapy for infectious diseases such as blood infections and meningitis.

9. See Susskind, supra note 5, at 174. Two other well known medical expert systems are CADUCEUS (formerly INTERNIST) and CASNET. CADUCEUS performs diagnosis in the field of internal medicine. CASNET diagnoses, and advocates therapeutic measures for, the disease process of glaucoma, doing so in a fashion presumably that ophthalmologists have acclaimed to be akin to that of an expert in the field. See Shurkin, *Expert Systems: The Practical Face of Artificial Intelligence*, TECHNOLOGY REV., Nov. 1983, at 76.
that place a high value on knowledge of a single, well-defined subject.\textsuperscript{10} The current technology favors rule-oriented deductive processes and tends to shy away from intuitive, common sense reasoning.\textsuperscript{11} The key step in expert system development is trial-and-error consultation between expert and programmer to develop the rules and exceptions that human experts learn from theory and practice.\textsuperscript{12} This process has become known as "knowledge engineering."

\section*{II. APPLYING CURRENT LEGAL PRINCIPLES TO EXPERT SYSTEMS}
\subsection*{A. INTRODUCTION}

Courts have not yet addressed physical and economic injury due to expert system errors in the tort liability or contractual context. In the tort liability area, the issue of whether the expert system developer should be subject to the standard of negligence or strict liability depends on whether the expert system in question is a product or a service. If an expert system is a product, strict liability in tort principles may apply. If an expert system is a service, negligence principles should be used. The easy answers are the extremes: a mass produced, mass marketed expert system would receive strict liability in tort treatment; whereas, a custom made expert system developed specifically for one user would most likely be subject to a negligence standard.

An expert system falling between the above extremes provides for a more difficult analysis in both the tort liability and contractual contexts. Because of the complexity and comparison of human reasoning to expert systems, in most cases, the courts will have difficulty in classifying expert systems as products or services. The choice will probably

\textsuperscript{10} Nycum & Fong, \textit{Artificial Intelligence & Certain Resulting Legal Issues}, 6 U.S.C. COMPUTER L. INST. § 4, at 4-010 (May 9, 1985) (available at the University of Southern California Law Library).

\textsuperscript{11} Expert system developers currently struggle with the concept of programming intuitive and "common sense" reasoning. This is based largely on the fact that physical and social scientists offer no "rule based" explanation of how the human mind operates in its intuitive and "common sense" manner.

\textsuperscript{12} An example of a MYCIN inference rule is:

\begin{verbatim}
IF (1) the infection is primary-bacteremia, and
(2) the site of the culture is a sterile site, and
(3) the suspected portal of entry of the organism is the gastro-intestinal tract,
THEN there is suggestive evidence (0.7) that the identity of the organism is bacteroides.
\end{verbatim}

Nycum & Fong, \textit{supra} note 10, at 4-008 (quoting Forsyth, \textit{The Expert Systems Phenomenon}, \textit{EXPERT SYSTEMS} 1, 6 (R. Forsyth ed. 1984)).
be based on policy with a close regard to analogous case law involving products and services similar to expert systems.

The policies which courts should consider, include: (1) the overall importance of new development of expert systems; (2) the specific societal gain from the use of the expert system in question; (3) the specific harm to the plaintiff; and (4) the ability of the expert system industry to self-regulate the competence and skill of those involved in the development of expert systems.

B. EXPERT SYSTEMS: PRODUCTS OR SERVICES?

Current tort and contract analysis require that expert systems must be divided into two camps: (1) products (typically mass produced and distributed programs) and (2) services (typically customized programs with unique characteristics).

Typically, products are manufactured items like autos or soda bottles,\textsuperscript{13} produced by companies in the business of the production, marketing, and distribution of such items.\textsuperscript{14} Services are functions performed by professionals such as architects, lawyers, doctors, dentists, consultants and engineers. Providers of services are typically not held to a strict liability standard.\textsuperscript{15} An expert system that is mass produced and distributed but operates effectively to perform the service normally offered by a human expert clouds the distinction between product and service. In addition to the element of the human expert comparison, expert systems requiring human intervention in the program's deductive process renders the classification as either product or service even more difficult. One resolution may be to look at that which the expert system provides. If the characteristics of the expert system take the form of a service for which a human expert may be employed to perform,\textsuperscript{16} then that expert system should arguably be designated a service.

\textsuperscript{13} Exploding soda bottles have been considered eligible for products for strict liability since the earliest product liability cases. See Escola v. Coca Cola Bottling Co., 24 Cal.2d 453, 150 P.2d 436 (1944).


\textsuperscript{16} Example: An expert system which instructs the user that, based on the user's investment portfolio, it would be wise to invest in municipal bonds, takes the form of a service since one would employ a human investment expert to render such a service. If this particular "investment advice expert system" made a mistake in its analysis similar to a mistake a human expert could conceivably make, the expert system developer should not be held to a stricter standard of liability than the human investment expert.
1. Uniform Commercial Code

A fundamental issue in resolving a contract dispute involving a software transaction is whether Article 2 (hereinafter "Article 2") of the Uniform Commercial Code (hereinafter the "U.C.C.") or the common law governs. For purposes of contract law, section 2-105(1) of the U.C.C. defines "goods as all things . . . which are moveable at the time of identification to contract for sale . . . ." The judicial treatment of software is generally consistent with the judicial treatment of commercial transactions. The conclusion that a contract to develop custom software is a service contract outside Article 2, while the sale of standardized software is within Article 2, is analogous to the same distinction made by courts between custom and standard goods in general.

In most reported cases involving software contract issues, the status of the software contract as a sale of goods under the U.C.C. has been stipulated by the parties or not substantially contested. In a few cases involving the sale of software, Article 2 was applied to the transaction. For example, in *RRX Industries, Inc. v. Lab-Con, Inc.*, the court held the California version of the U.C.C. to be applicable to a contract for the purchase of software without any discussion of whether the software was a good. The main thrust of the court's analysis, rather, was its finding that the sale of the software with accompanying services was within Article 2.

The *RRX* court relied on the "predominant feature" test to reach its conclusion. Under this test a contract involving both aspects of sale and service is classified according to the predominant aspect. The court concluded that because the seller's contractual obligations to install the software, to repair any software errors, and to train the buyer's employees in the operation of the software were merely incidental services, the transaction fell within Article 2.

Similarly, in *Compu-Med Systems, Inc. v. Cincom Systems, Inc.*, the court applied Article 2, under Ohio law, to the transaction without addressing the threshold question of whether software was a good. The cursory nature of the analyses in *RRX Industries* and *Compu-Med* indi-
icates little judicial hesitancy with the conclusion that software is a good under Article 2.

The courts similarly treat the sale of software as goods under Article 2 where a single agreement exists between the parties involving the sale of hardware and software. In Triangle Underwriters, Inc. v. Honeywell, Inc., the sale of a computer system consisting of hardware, standard software, and custom software resulted in breach of contract when the system failed to function properly because the software did not operate as promised. The court concluded that the transaction involved the sale of goods under Article 2.

In another case, Dreier Co., Inc. v. Unitronix Corp., the court concluded that:

It is clear that the sale of a computer system consists not only of physical goods, but of substantial services essential in producing the final product. Nevertheless, most authorities agree that the sale of a computer system involving both hardware and software is a "sale of goods" notwithstanding the incidental service aspects of the sale. Therefore, Article 2 of the Uniform Commercial Code applies.

Although courts have found Article 2 applicable to most sales of hardware and software under a single agreement, less consistent results exist when the computer hardware and software are leased rather than sold.

Thus, in most cases, a contract to purchase an expert system will be subject to Article 2 based on the judiciary's lack of hesitancy to conclude or find otherwise. In the event that an expert system developer contracts with a user to develop an expert system and the contract's predominant feature is to provide programming services, then the applicability of Article 2 is less apparent. An expert system developer who

26. Id.; but cf., Samuel Black Co. v. Borroughs Corp., 33 U.C.C. Rep. Serv. (Callaghan) 954, 962 (D.Mass. 1981)(court unsure whether the computer transaction was covered by Article 2, but applied it anyway since the outcome was the same under both common law and the U.C.C.).
28. Id. at 267.
29. See, e.g., Matter of Community Medical Center, 623 F.2d 864 (3d Cir. 1980)(where the court concluded that the lease was not the equivalent of a sale since the lessee could not purchase the computer system in question at the end of the lease term); O J & Co. v. General Hospital Leasing, Inc., 578 S.W. 2d 877, 878 (Tex. Civ. App. 1979)(where court applying Texas law concluded that Article 2 did not apply since the computer hardware and software were leased and not sold).
desires to enter into a development contract which would fall within Article 2, therefore, should draft the contract with language portraying the transaction as sale of a computer program rather than as a contract to obtain programming services.

2. **Tort Law**

Under traditional tort law, the labelling by a court of an expert system as a product or service will determine whether the negligence or strict liability in tort standard should apply. Under strict liability in tort law (hereinafter "strict liability"), the policies underlining the emergence of strict liability make a strong case for treating mass marketed expert systems as a product. Where a program is distributed on a mass market basis, the analogy to any other product is clear. Most completed programs distributed to more than a single user possess sufficient characteristics of a product to fall under strict liability principles. In such cases, the individuals who are functioning within the foreseeable risk zones rely on the expert system's safe structure and design, and the manufacturer and designer can spread the risk of losses through pricing and insurance. Furthermore, the victim's ability to trace defects to the programming operation and to establish negligence in that remote enterprise is rendered difficult or impossible by the context. The expert system is the embodiment of complex programming techniques which require input of expert information into the knowledge base and the structure of the knowledge base itself. It is only when the output of the expert system is restricted to a single user that the programmer's undertaking lacks the affirmative commitment of a product to the market place that establishes potential liability for defects.\(^{30}\)

Classification as an information generator rather than as a traditional form of goods does not affect the issue of whether the expert system is labelled as a product to which strict liability principles. A strict liability analysis was applied to an "information product" in the Ninth Circuit case, *Aetna Casualty & Security v. Jeppeson & Co.*\(^{31}\) *Jeppeson* involved instrument approach charts distributed for pilots making instrument approaches to various airports. A defect in a chart allegedly caused the crash. The court concluded that the charts were products for purposes of applying principles of strict liability. The analogy to software products and expert systems is relatively clear, assuming a clear connection between a risk of physical injury and a defective product.

Nevertheless, the operation of some expert systems may demand intervention by a human user during the deduction process. In many

---

31. 642 F.2d 339 (9th Cir. 1981).
applications, the user and expert system may join to create a diagnosis. For example, in *Jeppeson*, the court held that the pilot was negligent in his reliance on the defective charts. In such an area, where physical injury is at risk, it is improbable that an expert system developer would be held to a strict liability standard where the physical injury was caused by the output of the expert system and the expert system required significant intervention and review by the professional using the program.

Imagine an engineer who uses an expert system to design a bridge. The knowledge base of the expert system contains all the necessary mechanics for applying the physical laws to building the bridge. During the process of its configuration and calculation, the expert system stops and requests that the user-engineer review some of its deductions and directs the user-engineer to approve before proceeding. At a different point in the program, the expert system requires the user-engineer to input important cost, weather, and environmental information. The expert system makes some final adjustments and produces a plan for the bridge. The user-engineer implements the plan. Two weeks after the bridge is built, it collapses. It is proven that the defective plans caused the collapse. Under this scenario, it appears that the expert system described is not a product for strict liability purposes, because the actual product, the design plans, is a product of the joint effort of the expert system and the user-engineer. In this case, the expert system functioned as a service for the engineer and required important user intervention (i.e., cost, weather, and environmental information input and verification of calculations). Even if it were widely distributed and mass produced, the interaction with the professional user makes it appear more like a service. On the other hand, it is conceivable that many expert systems may replace the professional entirely and require no coupling of effort to reach a diagnosis, decision, or plan.

Consider the following example, which is similar in effect to the fact situation in *Jeppeson*. A commercial airline installs a computer which uses an expert system to assist pilots in avoiding mid-air collisions. The program is valuable to the airline because it provides warnings and data sufficient to assist in making a "split-second" decision to avoid other aircraft. The program does not pause its analysis to request human intervention or require verification. The pilot/air traffic controller acts on the output or advice of the program, and a collision ensues. In this case, the expert system looks more like a product than the previous example, because the expert system in this instance provides a

32. *Id.* at 343.
33. *See supra* text accompanying notes 30-32.
product (information) which is expected to be relied upon without verification or human intervention.

Therefore, whether a mass produced and distributed expert system is a product for strict liability purposes depends on the actual make-up and orientation of the expert system to a particular area. A developer which desires to avoid classification of its expert system as a product should be advised to design and market the expert system that requires some human intervention in reviewing the analysis of the expert system.

An argument for classifying an expert system as a service is that it can be individually tailored to perform a specific task. Viewing the interaction between the programmer and the user in the development of a specific expert system as a service, strict liability would not be applicable to such a one-of-a-kind product.\footnote{34. See Comment, Computer Software and Strict Liability, 20 SAN DIEGO L. REV. 439, 444 n.17 (1983) (arguing, on balance, for classification of computer programs as products for strict liability purposes).}

Strict liability might apply to expert systems (deemed to be "products") that cause certain types of injury involving ordinary programming errors, which are likely to be considered production defects, as well as to those with design defects that cause injury, especially when the defective expert system directly operates a machine that inflicts injury on an individual (a la RoboCop).\footnote{35. See Brannigan & Dayhoff, Liability for Personal Injuries Caused by Defective Medical Computer Programs, 7 AMERICAN JOURNAL OF LAW & MED. 123 (1981).} If on the other hand, the injury is related to a conscious design choice, the defendants are likely to be judged according to a negligence standard.\footnote{36. Id. at 131.}

Expert systems that are mass produced which cause injuries without any required human interaction with the expert system would presumably be classified as a product for purposes of strict liability law. Many expert systems will not take this form. Thus, courts will have the difficult task of deciding whether an individual expert system is a product or a service.

The following elements of an expert system will make the distinction between a product and a service more difficult to make: (1) one-of-a-kind applications; (2) human intervention as part of the expert system's deductive process; (3) an application area where no reasonable user would blindly rely on the output of the expert system; (4) experimental programs where the user is aware of the infancy of the testing process; and (5) where the user has contracted a programmer to develop a expert system and compensation is for the programmer's services rather than the value of the expert system program.
C. RECOVERY UNDER THE UNIFORM COMMERCIAL CODE

As already discussed, most contracts for expert system procurement will likely be held, or stipulated to by the parties, as contracts for a product under the U.C.C. Generally, courts have had little difficulty in concluding that software procured alone or in connection with computer hardware is subject to the provisions of Article 2.

1. Statute of Limitations

The U.C.C. limits contract actions to four years after the cause of action arises, and allows parties to decrease this time to as little as one year. Vendors usually insist upon a one-year limitations period for most claims except for nonpayment because of a perception that if a dispute arises over a computer vendor's performance, the user probably will not commence litigation until long after the original breach by the vendor occurs.

In Int'l Business Mach. v. Catamore Enterprises, Inc. [548 F.2d 1065 (1st Cir. 1976), cert. denied, 431 U.S. 960 (1977)], which involved numerous transactions between IBM and a jewelry company in Rhode Island, a one-year limitations period contained in a written document was applied to an earlier oral agreement even though the court refrained from deciding whether it had been superseded in its entirety by the written agreement. Instead of suing IBM, Catamore withheld payment. The limitations clause in question provided that '[n]o action, regardless of form, arising out of the services under this Agreement, may be brought by either party more than one year after the cause of action has accrued [sic], except that an action for nonpayment may be brought within one year of the date of last payment.' Since the alleged breach by IBM occurred well prior to the date of last payment, IBM was able to rely on this clause and sue Catamore for failure to pay, while Catamore was unable to counterclaim for IBM's breach.

The statute of limitations problem as applied to expert systems, becomes a barrier to many users where the vendor has incorporated into the agreement that neither party can bring a cause of action more than one year after the cause of action arises. The complexity of an expert system and its relation to the occurring harm may be so obscure and uncertain at the time that the cause of action arises, that it may take the user more than the one year allowed under the agreement to discover

37. See supra text accompanying notes 16-29.
38. See supra text accompanying notes 16-29.
that the vendor is, in fact, in breach of the terms of the contract. If
every time a possible breach occurs in connection with the agreement
for procurement of an expert system the user sues the vendor\textsuperscript{42}, it may
be difficult to state facts sufficient to support a cause of action.

2. Integration or Merger Clauses

During the stage of negotiations for the procurement of an expert
system, the user will typically be subjected to a wide range of represen-
tations as to the capabilities of the company’s software and its perform-
ance capabilities on various computer hardware configurations. “Sales
personnel . . . may make oral representations regarding commitments
by the vendor to program the software correctly, maintain the equip-
ment, respond to service calls with a fixed number of hours, replace de-
fective software, and perform other services, all designed to convince
the user that the overall package . . . is the best that money can buy.”\textsuperscript{43}
The user makes his choice based on the combination of the written
materials offered, the oral representations made, and deference to
others in the same trade.

Almost invariably, however, most computer contracts (usually ven-
dor standard form contracts) have a clause providing that there are no
understandings or agreements exist between the parties except as speci-
fied in the written contract, and that the vendor has no obligations to
the user except as expressly set forth in the contract. So-called merger
or integration clauses are generally held to be valid.\textsuperscript{44} Usually, the ex-
clusion of evidence with regard to contract claims cannot be avoided
and the plaintiff must therefore pursue tort claims in order to introduce
the damaging evidence.

In Applications, Inc. \textit{v. Hewlett-Packard Corp.} . . . such evidence was
held to be admissible to prove fraud claims. The court concluded with
regard to warranties, the written instrument was the complete and fi-
nal embodiment of the parties’ agreement and that no evidence of
other alleged oral warranties or representations could be introduced
for the purpose of demonstrating that certain critical warranties had
been made.\textsuperscript{45}

Obviously, a user would be well-advised to consider the parol evi-
dence rule during negotiations by insisting that all the understandings

\textsuperscript{42} The user may sue whenever anything goes wrong related to the expert system so
as to preserve a cause of action against other parties.
\textsuperscript{43} Bernacchi, Davidson, & Grogan, \textit{supra} note 41, at 405.
\textsuperscript{44} Bernacchi, Davidson, & Grogan, \textit{supra} note 41, at 405.
Data Systems, 708 F.2d 385 (1983) (where the court held that specifications of computer
response time on which buyer relied were express warranties that could not be quashed
by a disclaimer in the contract provisions).
of the parties be incorporated into the agreement. This practice may prove extremely difficult when the potential user considers buying an expert system from a vendor licensed to sell the program on behalf of a manufacturer. Typically, the vendor must abide by the limited warranty and disclaimer form supplied by the manufacturer. In most cases of the procurement of a mass produced expert system, it will be virtually impossible to negotiate warranty terms. Only where both parties are sophisticated and relatively equal in bargaining power, may the merger or integration clause depict what the parties actually bargained.

The inability for many buyers to incorporate representations made in reference to software, either contractually or by proving such representations, is a problem which may spawn legislation implying warranties in computer advertising. Such a bill has been introduced before the California State Legislature. The bill, Assembly Bill 1507 (hereinafter "AB 1507"), as amended by Assemblywoman Glorina Molina, would hold manufacturers and retailers liable for their advertising claims for a product.

AB 1507 provides that every sale or lease of new computer products or parts is to be accompanied by an implied warranty of merchantability if sold by a dealer, by an implied warranty of fitness where the manufacturer or seller has reason to know of the required purpose and that the buyer is relying on the skill or judgment of the manufacturer seller, and an express warranty that the product conforms to promises and affirmations in the manufacturer or seller's advertisements. AB 1507 pro-

46. Hardware and software warranty and liability disclaimers often resemble the following disclaimer drafted by the manufacturer of a popular computer system:

**DISCLAIMER OF ALL WARRANTIES AND LIABILITY**

--- Inc. and --- make no warranties, either express or implied, with respect to this manual or with respect to the software described in this manual, its quality, performance, merchantability, or fitness for any particular purpose. --- computer software is sold or licensed "as is". The entire risk as to its quality and performance is with the buyer. Should the program prove defective following their purchase, the buyer (and not --- Computer, Inc., or ---, their distributors, or their retailers) assumes the entire cost of all necessary servicing, repair, or correction and any incidental or consequential damages. In no event will --- Inc., or --- be liable for direct, indirect, incidental, or consequential damages resulting from any defect in the software, even if they have been advised of the possibility of such damages. Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation exclusion may not apply to you.


47. Assemblywoman Molina got the idea for the bill after she purchased an Apple Lisa computer to do mass-mailing projects for her office. After six months of trying to get the computer to do what she had bought it for, Molina got Apple to replace a microchip unit in her Lisa that, in effect, converted it to a less expensive Macintosh. L.A. Daily J., August 8, 1985, at 2, col. 1.
hibits the disclaimer or limitation of duration of those warranties and the recovery of attorney’s fees if the buyer prevails in an action to enforce liability under AB 1507, as enacted. Finally, AB 1507 states that the remedies provided to the buyer are intended to be cumulative in that all procedures, rights, or remedies are still available under any other provision of law.

To date, no such legislation exists. Thus, to help alleviate the effect of the integration clause, a user should insist that any sales brochures or sales documents, including any responses from the computer vendor to the user’s request for proposals, should be attached to, and expressly made a part of, the agreement. With these documents attached to the formal agreement, the user will ensure that the representations and statements of the vendor’s sales representatives which induced the user to procure the computer systems will not be excluded from consideration in any interpretation of the agreement during a dispute. Vendor’s should be careful about their advertisements and conduct of sales representatives such that they describe carefully their product in the best light possible without misleading a potential buyer.48

3. Disclaimer of Warranties

If the U.C.C. applies to expert system software, damages for breach of express warranty include the difference in value between the defective and warranted software,49 plus incidental and consequential damages.50 Express warranties usually appear with warranty disclaimers and limitations on remedies.51 Generally, the U.C.C. permits these limitations.52 The U.C.C. provides two important implied warranties: the implied warranty of merchantability and the implied warranty of fitness for a particular purpose.53

In order for the implied warranty of merchantability to apply, there must be a “contract for [the] sale . . . [of] goods” and the seller must be a merchant “with respect to goods of that kind.”54 The implied warranty of merchantability imposes on the merchant of goods an obligation to provide goods which are merchantable, which requires the goods to pass without objection in the trade, and to be fit for the ordi-

48. Bernacchi, Davidson, & Grogan, supra note 41, at 411-12.
50. Id. at §§ 2-714(3), 2-715.
53. Id. at § 2-314.
54. Id. at § 2-315.
55. Id. at § 2-314(1).
mány purposes for which they are used. An implied warranty of 
merchantability may be disclaimed either orally or in writing, if the 
writing is conspicuous, and as long as the word merchantability is used 
in the written or oral disclaimer.

In order for the implied warranty of fitness for a particular purpose 
to apply, the vendor must "at the time of contracting [have] reason to 
know any particular purpose for which the goods are required and that 
the buyer is relying on the [vendor]'s skill or judgment to select or furn-
ish suitable goods ...." This situation will almost always be present 
in most sales of expert systems. If MYCIN is purchased by a doctor, it 
is obvious to all concerned that the doctor is going to use it for bacterial 
analysis.

An unsophisticated user, with little or no knowledge of computers, is 
approached by sales representatives of a computer company who evalu-
ate the user's business and make recommendations for a system which 
will meet those needs. A more sophisticated user may create a general 
list of its requirements, may incorporate it into a 'request for propos-
als,' and may distribute this document to suitable computer vendors. 
Each vendor responds by indicating how its particular system with ap-
propriate modifications to its software will meet the user's needs. Re-
lying on the expertise of the computer company, the user agrees to buy 
the system, confident that the system will suit its needs.

In this context it has been recognized that the warranty of fitness 
for a particular purpose is quite appropriate. For example, in Sperry 
Rand Corp. v. Industrial Supply Corp., "the implied warranty of fit-
ness for a particular purpose was 'clearly' established when the vendor 
sent the user a detailed set of written recommendations based upon the 
vendor's study of the user's needs."

The warranty of fitness for a particular purpose is also typically 
"disclaimed by vendor form contracts which usually contain language 
similar to the following: 'THERE ARE NO UNDERSTANDINGS, 
AGREEMENTS, REPRESENTATIONS OR WARRANTIES, EX-
PRESS OR IMPLIED (INCLUDING ANY REGARDING 
MERCHANTABILITY OR FITNESS FOR A PARTICULAR PUR-

56. Id. at § 2-314(2). Other requirements also need to be fulfilled, but the require-
ments mentioned in the text are more appropriately applied to computer related contracts 
than are the others. See Bernacchi, Davidson, & Grogan, supra note 41, at 412.
57. U.C.C. § 2-316(2) at comment 3 (1978).
58. Id. at § 2-315.
59. See supra text accompanying notes 8-9.
60. Bernacchi, Davidson, & Grogan, supra note 41, at 412-13.
61. Id. at 413. See e.g., National Cash Register Co. v. Adell Indus., Inc., 57 Mich.App. 
413, 225 N.W. 2d 785 (1975).
62. 337 F.2d. 363 (5th Cir. 1964).
63. Bernacchi, Davidson, & Grogan, supra note 41, at 413.
POSE) NOT SPECIFIED HEREIN, RESPECTING THIS CONTRACT OR THE EQUIPMENT HEREUNDER." The U.C.C. provides that an implied warranty of fitness for a particular purpose may be excluded by general language if that language is in writing and is conspicuous.

Even in the absence of an express exclusionary clause, implied warranties may be excluded in several other circumstances. If the user examined or had an opportunity to examine the expert system before entering into the contract, the implied warranties do not apply to any defect that an examination might have revealed.

4. Limitation of Damages

Absent contractual modification, the U.C.C. provides that the buyer in most cases may recover consequential damages arising from any breach of contract. "Consequential damages" include "any loss resulting from general or particular requirements and needs of which the seller at the time of contracting had reason to know . . . and injury to person or property proximately resulting from any breach of warranty." In general, consequential damages encompass the effects resulting from the breach, such as lost profits or unrealized but anticipated personnel savings. The U.C.C. provides that "consequential damages may be limited or excluded unless the limitation . . . is unconscionable." Vendors typically incorporate into the agreement a clause which places a ceiling on liability, usually by limiting the amount paid by the user under the contract and by excluding liability for "indirect, special or consequential damages."

5. Unconscionability

The U.C.C. provides that a court may refuse to enforce any contract or clause of a contract on a finding that the contract or clause was un-

---

64. Bernacchi, Davidson, & Grogan, supra note 41, at 414 (quoting Investors Premium Corp. v. Burroughs Co., 389 F.Sup. 39, 45 (D.S.C. 1974)).
65. U.C.C. § 2-316(2) at comment 4 (1978).
66. Id. at § 2-316(3)(b). This "examination" should be clearly distinguished from "inspection" or "acceptance testing," which occurs after the contract has been entered into, but before the expert system has been accepted as conforming to the contract. Id. at § 2-316, Comment 8. In contrast, examination is intended to refer to the user's selection process before entering into an agreement. Id. An example relevant to expert systems sales would be inspecting programs and watching demonstrations of a particular vendor's expert system prior to signing a purchase agreement.
67. Id. at § 2-715(2).
70. Teamsters Security Fund v. Sperry Rand Corp., 6 C.L.R.S. 951, 968 (N.D. Cal. 1977). Such clauses also include "incidental" and "exemplary" damages, "lost profits" and "other pecuniary loss," or a mixture of these phrases.
conscionable at the time it was made.\textsuperscript{71} The purpose of this doctrine is to prevent "oppression and unfair surprise and not [to disturb an] allocation of risks because of superior bargaining power."\textsuperscript{72} A clause or contract normally is not refused enforcement unless both procedural and substantive unconscionability are present.\textsuperscript{73}

In The Glovartorium, Inc., v. Nat'l Cash Registar Corp.,\textsuperscript{74} the buyer relied on a model and the seller's assurances which did not in fact reflect the true performance limitations of the business records system computer program. The lower court indicated that the limitations language was unconscionable in the face of this clear reliance by an inexperienced buyer. The appellate court affirmed a substantial damage award based on the lower court's finding of fraud, however, without expressly affirming the unconscionability claim.

"The argument is that even an otherwise sophisticated business person can not reasonably appreciate the significance of warranty and damage limitation language in this context. This argument has been generally rejected . . . ."\textsuperscript{75} As one court noted, businessmen are presumed to act at arm's length and mere technological naiveté is inadequate cause to disrupt this presumption on issues relating to assumption of risk.\textsuperscript{76} Although a businessman may not be sophisticated in computers, the presumed sophistication in commercial matters coupled with an ability to obtain legal and other assistance is ordinarily decisive.\textsuperscript{77}

The application of unconscionability to expert systems will probably center on the business sophistication of the user. Currently, most expert system users are highly trained professionals. Nothing appears special about this group to require a different application of the unconscionability standard.

D. RECOVERY UNDER CURRENT TORT PRINCIPLES

"Injury caused by an expert system can be traced to programmers, users, distributors or manufacturers."\textsuperscript{78} For example, an expert system

\textsuperscript{71} U.C.C. § 2-302 (1978).
\textsuperscript{72} Id. § 2-302 at Comment 1.
\textsuperscript{73} Procedural unconscionability focuses on the use of pressure or deceptive tactics, the presence of extreme bargaining inequality, and significant differences in the sophistication of the parties. Substantive unconscionability encompasses terms that are unreasonably favorable to one party. Nimmer, supra note 30, ch. 6 at 52. See also Leff, Unconscionability and the Code - The Emperor's New Clause, 115 U.P.L. REV. 485, 487 (1967); White & Summers, Handbook of the Law Under the Uniform Commercial Code 47 (2d ed. 1980).
\textsuperscript{74} 684 F.2d 658 (9th Cir. 1983).
\textsuperscript{75} Nimmer, supra note 30, ch. 6, at 53.
\textsuperscript{76} Earman Oil Co., Inc. v. Burroughs Corp., 625 F.2d 1291, 1300 (5th Cir. 1980).
\textsuperscript{78} Nycum & Fong, supra note 10, at 20.
that monitors air traffic control at a busy airport and provides advice to the air traffic controllers can fail and cause injury due to any combination of four sources of error: (1) errors in the program;\(^79\) errors in the knowledge base;\(^80\) (3) unintended use or failure to use or undue reliance by the air traffic controller;\(^81\) and (4) hardware failure. The more difficult issue arises in determining the standard of reasonableness to apply to the air traffic controller on relying or not relying on the expert system. As expert systems are used more, one would be negligent not to use an expert system in that position.

1. **Proof of Negligence**

   Since expert systems are developed by the combined efforts of programmers and experts in a particular field, it may be extremely difficult to actually determine who is responsible for a design defect. The human expert may have made an error explaining to the programmer how a particular situation should be handled. Or, the programmer could have developed a poorly fitted control structure and failed to encode the human expert’s direction properly into the knowledge base. In the air traffic control example, if a plane crashed as a result of the pilot’s reliance on the advice given by the expert system, the plaintiff would be faced with an enormous task of proving that the expert system failed. In a negligence action, where the burden is on the plaintiff to prove that the defendant breached his duty of care (unless the doctrine of *res ipsa loquitur* could be used to shift the burden),\(^82\) it would take a substantial team of professionals to look into the matter and determine the proximate cause of the error.\(^83\)

   If the air traffic controller could prove that he was acting reasonably by relying on the expert system because it would have been unsafe not to do so and it was customary for all air traffic controllers to act in the same way, the ultimate victims would be hard pressed for recovery. The air traffic controller could put on terrific evidence showing that air

---

\(^79\) Errors in the program may be a logic error (i.e., incorrect encoding of a programming step) or a design error (i.e., inefficient or incorrect choice of a programming step).

\(^80\) Errors in the knowledge base could mean inappropriate data or an erroneous inference rule which escaped testing. Nycum & Fong, *supra* note 10, at 21.

\(^81\) “Undue trust is placed on machines by some people who are either unaware of a machine’s capabilities or who purposefully wish to deflect the opinions of others. While such cases will undoubtedly arise, it is my belief that they will be infrequent, since people will seek to comprehend and question the results of computing machines as they have done for other human artifacts.” Nycum & Fong, *supra* note 10, at 21, n.39 (citing M. DERTOUZOUS, THE INFORMATION REVOLUTION, 5 (1983)).


\(^83\) Dertouzous, however, reminds us that “[a] jumbo aircraft is equally complex, yet we make sure that it is comprehensible and that we know whom to prosecute if it crashes.” DERTOUZOUS, *supra* note 81, at 15.
safety is greatly enhanced by the advent of expert systems and that the expert systems help air traffic controllers increase safety in ways not previously possible.

2. Standard of Reasonable Care

Another problem facing the potential tort plaintiff is proving that the defendant (when identifiable) breached a duty of care owed to the plaintiff. The general standard of care is usually stated as that of the “reasonable man of ordinary prudence.”\(^4\) This standard is difficult to establish, however, since the industry has so few standards by which to measure performance. One possible approach is to hold programmers and computer consultants to a standard of a professional, and thus require a higher standard of care. This would ease the plaintiff’s burden of proof for a showing of negligence. Such a claim has been labeled as computer malpractice.\(^5\)

A computer malpractice claim was given short shrift in Chatlos Systems v. Nat’l Cash Register Corp.\(^6\) The trial court dismissed Chatlos’ computer malpractice claim in the first footnote of its opinion. The court found that the technical complexity and relative importance to the business community of selling and servicing computer systems was insufficient justification to impose greater potential liability.\(^7\) The case was decided on contractual grounds in favor of Chatlos. The court may have confronted the issue directly had the tort been the only grounds of recovery.

If a plaintiff were successful by proving that an expert system was the cause of his injury, and managed to identify those responsible for the cause, one of the accused individuals might be a professional (besides the programmer) to which malpractice, or an elevated standard of care, would ordinarily apply. For example, if the expert system in question concerned architecture, and the plaintiff proved that the architect, upon which the expertise of the program was based, would have been guilty of malpractice had the information come directly from him,

\(^{84}\) See W. Prosser, supra note 82, § 32, at 150.


\(^{86}\) 479 F.Supp. 738, 740-41 (D.N.J. 1979), modified 635 F.2d 1081 (3d. Cir. 1980), cert. denied, 457 U.S. 1112 (1982) (Plaintiff, Chatlos, was involved in the design and manufacture of cable pressurization equipment for the telecommunications industry. Defendant NCR, designed, manufactured and sold computer systems, programs and services. Chatlos signed a systems services agreement for the sale of a NCR computer system to be operational within six months. The system had problems from the time it was installed; attempts were made to correct the system until a certain time when relations between the parties ceased.)

\(^{87}\) Id.
should the architect be liable in tort? The fact that his knowledge is
encoded into a computer program clouds the analysis. If this expert sys-
tem were sold and advertised as the replacement of the architect’s serv-
ices, the reliance issues under traditional malpractice causes of action
would be present.

3. **Strict Liability in Tort**

If an expert system is deemed a product, defective and unreasona-
bly dangerous, recovery under strict liability principles may be applied
even if reasonable care were used or if no amount of care could have
prevented the injury. The appellate courts have not yet dealt with the
issue of whether a strict liability claim can be applied to computer
software. Such a claim will present a difficult legal analysis further ag-
gravated by the technical complexities.

**III. EFFECT OF LAW ON EXPERT SYSTEM DEVELOPMENT**

At present, plausible legal theories for recovery from an injury
cau sed by an expert system include (1) breach of warranty and breach
of representations, (2) breach of contract, (3) ordinary negligence,
(4) professional negligence (malpractice), and (5) strict liability in tort.

A. **Breach of Warranty, Breach of Representations and
Breach of Contract**

Breach of warranty, breach of representations and breach of con-
tract causes of action do not pose any threat to expert system develop-
ment. Both parties to the contract can employ basic contract principles
for protection. The manufacturer or distributor of an expert system can
utilize the numerous disclaimers available to minimize losses. The pur-
chaser or licensee of an expert system can negotiate for important pro-
tection. No inherent characteristics to expert systems merit different
treatment under contract law. In most circumstances, parties to the
contract do not lack sophistication. In the case of “mass produced, mass
marketed” expert systems, sales of such expert systems will fall within
Article 2 of the U.C.C. and provide the unsophisticated user/purchaser
whatever protection each state desires to give to such entities under its
applicable commercial code.

B. **Ordinary Negligence**

Difficulties exist in establishing an appropriate standard of reasona-

---

89. *Cf.* Walker, *Strict Liability for Software Programs*, 27 *FOR THE DEFENSE* 14, Octo-
ber 1985. *See also supra* text accompanying notes 21-25.
bleness for expert system developers because of the infancy of expert system programming techniques and procedures. A key determinant in expert system development is what standard will be applied to developers. To date, courts have not recognized programmers in the same class as other professionals operating under a higher level of care. As expert systems become more important to society, the law may adopt a new standard of care for computer programmers.

One problem with expert systems is determining who ought to be consulted as to the standard of care. The determination of the standard of care might depend on where the mishap arose (i.e., in the design of the program, in the knowledge base, or in the interaction with the user.) This affects the determination of which different groups should be considered “knowledgeable” and capable of establishing a standard. Thus, if the cause of injury is traced to a faulty knowledge base, then the “reasonable” standard of care should be established by the ordinary reasonable care of an average knowledge base engineer in the industry.

Negligence, as a cause of action, forces the parties to a human standard. This means that the developer is not forced to be a perfect insuror or a perfect programmer, nor is the user permitted to be naive, unwarrantly trusting or willfully blind to the potential hazards foregoing the use of his own judgment. The human standard will ensure that the parties act reasonably under the circumstances. Even if the user/plaintiff has followed instructions and his actions were logical, and if an injury is caused by the output or action of an expert system, the defendant can still escape liability under a negligence claim if he acted reasonably.

The advent of a new technology such as expert systems brings great societal gains while often imposing costs on certain individuals. Who should bear such costs? Perhaps the user/plaintiff should bear some of the costs to the extent it would bear the costs under similar services it would have received from a human expert. The user/plaintiff should expect that, in its present state, expert system technology is no more than the attempted emulation of human experts. The court should treat the situation as it would had the plaintiff relied on a human expert rendering the same service. As long as the expert system performs as reasonable as its equivalent human experts, an injured plaintiff will not recover under an ordinary negligence cause of action.

C. NEGLIGENCE OF A PROFESSIONAL

A computer programmer is presently required to act only as an average reasonable person under similar circumstances. The failure of

90. See supra text accompanying notes 82-87.
91. See W. Prosser, supra note 82, at 143, 150.
the courts to allow a computer malpractice claim\textsuperscript{92} may give a programmer, possessing superior intellectual capacity, an escape route when he really should be liable. If an expert system is marketed as the equivalent of or better than the human experts in a particular field, the programming should reflect the same — i.e., an expert system which performs like or better than a human expert.

Should the courts continue not to impose a higher standard of care, expert system developers will continue to have significant freedom to experiment and innovate with expert systems without concern for liability based on a yet undefined standard of care. Imposition of a higher standard of care would raise costs and limit entry to the market place. Holding a developer to a higher standard of care could result in more required education, computer malpractice insurance, and possible industry attempts to define programming standards. Such standards would certainly spawn more litigation because it would be easier to recover against developers held to a higher standard of care.

D. STRICT LIABILITY IN TORT

At present, no software programs causing injury have been held to a no-fault standard. Many commentators believe, however, that courts will have no difficulty in imposing strict liability on computer programs causing physical injury as long as the requisite components of a strict liability claim are present.\textsuperscript{93} From an expert system development perspective, imposing a strict liability standard may hamper society from benefiting from the developments of this important new technology.

Strict liability claims will first raise a primary definitional question, i.e., whether certain expert systems are products or services.\textsuperscript{94} As a result of the ambiguity of whether their expert system will be deemed a product or a service, many developers may alter the design, cost or features of their systems specifically in fear of being held liable based on a strict liability claim. The very nature of expert systems will raise inevitable situations where injury will occur as a result of the expert systems’ functions. A large class of “real world” problems are not susceptible to predictable, verifiable results. Expert systems try to emulate human reasoning; thus, an attempt to hold developers to a no-fault standard would require programmers to create a perfect electronic intelligence capable of out-performing a host of human experts.

\textsuperscript{92} See supra text accompanying notes 82-87.

\textsuperscript{93} See Gemignani, supra note 51, at 197; Comment, supra note 34, at 456; Brannigan & Dayhoff, supra note 35.

\textsuperscript{94} See supra text accompanying notes 12-16.
IV. PROPOSED LEGAL AND PROFESSIONAL STANDARDS

A. STRICT LIABILITY

The courts should not impose a strict liability standard on expert systems. Further, society must share the costs of developing this important new area of programming. Expert systems inherently are composed of fallible human reasoning which leads to non-verifiable results. Each user should carefully test a new expert system and rely upon its output only within the scope of verifiable results produced during the user's own testing process. If the manufacturer fails to inform the user of its expert systems' potential flaws, the user would be able to employ tort claims other than strict liability for recovery. Developers of expert systems should, however, be held to higher level of care when the expert system is advertised as a superior programming project.

B. LIABILITY OF PROFESSIONALS

Should the computer industry or government create a more defined and ascertainable class of expert system programmers, a computer malpractice claim should be available to potential plaintiffs. Such a claim would allow a user/plaintiff to bring a professional malpractice claim against the programmer/defendant. The plaintiff would have to prove that the defendant breached a graduated degree of care in the creation and implementation of the expert system above that of an average reasonable person of ordinary prudence.

The defendant could rebut by (i) demonstrating that the design and implementation of the expert system is consistent with the proven state of the art technology, and (ii) showing that he is qualified to participate in the development of an expert system by experience and/or education. The degree of care will be established similarly to that of lawyers and doctors in legal and medical malpractice suits, respectively.

C. HIGHER STANDARD OF CARE IN A NEGLIGENCE CAUSE OF ACTION

Courts or legislatures should impose a higher standard of care on expert system developers to elevate the standards of education and experience of the developers only if the industry is unable to regulate itself to an acceptable level of social responsibility.

The industry should require that expert system developers receive credentials showing that developers are competent to enter into important projects. With society becoming more dependent on computer related technology, society must require that the computer industry meets society's needs in a socially responsible manner.

If the computer industry fails to regulate itself, a government entity could impose licensing requirements for certain types of program-
ming including expert systems. Such licensing may include requisite education, mandatory internships, and both written and practical examinations.

V. CONCLUSION

Expert systems are an exciting innovation holding excellent potential for important societal benefits. The nature of expert systems, however, clouds legal analysis in many hypothetical circumstances because of the human reasoning comparison element.

The law will classify an expert system as a product or a service on a case by case analysis. Expert system developers will be confronted with the possibility of strict liability claims. Therefore, developers will initially modify or refrain from mass marketing expert systems for which their classification as a product or a service for purposes of tort and contract law application is unclear.

Imposition of a higher level of care on expert system programmers could be beneficial to society if the elevated legal standard of care causes the programming industry to regulate itself and insure users that the programming of an expert system meets a strict code of calibre.

*Tod M. Turley*