Some New Kinds of Authorship Made Possible by Computers and Some Intellectual Property Questions They Raise

Pamela Samuelson

Follow this and additional works at: https://scholarship.law.berkeley.edu/facpubs

Part of the Intellectual Property Law Commons

Recommended Citation

This Article is brought to you for free and open access by Berkeley Law Scholarship Repository. It has been accepted for inclusion in Faculty Scholarship by an authorized administrator of Berkeley Law Scholarship Repository. For more information, please contact jcera@law.berkeley.edu.
ESSAYS

SOME NEW KINDS OF AUTHORSHIP MADE POSSIBLE BY COMPUTERS AND SOME INTELLECTUAL PROPERTY QUESTIONS THEY RAISE

Pamela Samuelson*

Written texts have long had the potential to be powerful because of the effects they could produce on the actions of their readers. But writings have traditionally been understood to be incapable of doing work on their own. Since the invention of programmable digital computers, however, written texts have acquired the power to "utter work."1 Authors of these texts have been extending the meaning of authorship in ways previously unimaginable. This essay will discuss a few examples of the new kinds of authorship made possible by developments in computer technologies, and at least some of the questions such new works raise for the intellectual property law systems of the United States.

This essay will begin with a brief discussion of computer programming as a new kind of authorship and whether the act of creating computer language should be regarded as an act of authorship. It will go on

* Professor of Law, University of Pittsburgh School of Law. This essay was originally prepared as a paper for delivery at the Intellectual Property and Authorship Conference, held at Case Western Reserve University on April 19-21, 1991. I want to thank Bob Glushko, Wendy Gordon, Michael Joyce, James W. Moore and Elli Mylonas for their thoughtful comments and suggestions for revisions to an earlier version of the essay. I also want to thank Peter Jaszi and Martha Woodmansee for organizing the conference and giving me the opportunity to write this essay.


"Open sesame" and other magical incantations are examples from the world of literature of expressions that could directly cause things to happen. Such expressions are considered, at least by the mainstream culture, to be fictional. But apart from this, it is worth noting that only oral expressions of the incantations, not written ones, have been regarded as capable of producing direct results. Laws are probably the most powerful of written texts.

685
to discuss how some new kinds of electronic texts are changing how we perceive more conventional works of authorship, and the concept of authorship itself, as well as raising some challenging questions for intellectual property law.

I. COMPUTER PROGRAMS AND COMPUTER LANGUAGES

The first operable electronic digital computers were room-sized machines consisting of complex mazes of wires, switches, vacuum tubes and other hardware elements. When properly wired in a particular configuration and when given data on which to operate, such a machine could perform a sequence of operations that would produce a desired result, such as calculating the trajectory of a missile launched at $X$ angle, with $Y$ power and under $Z$ conditions. While one could change the values of $X$, $Y$ and $Z$ to calculate different trajectories, if one wanted the computer to solve a different equation, one had to reconfigure the machine by hand so that it could perform a different sequence of operations. To put it slightly differently, one “re-programmed” such a computer by disconnecting this wire from this terminus and reconnecting it to that terminus until the computer was re-wired for its new task.

The Hungarian mathematician John Von Neumann had the insight to perceive that one could build an electronic digital computer that would store not only the data on which the computer would act, but also a set of written instructions prescribing the exact order in which the computer should perform specific functions in order to complete a particular task. Von Neumann’s insight was a powerful one.


3. Occasionally, a moth or some other insect would short-circuit the computer’s operations by being in the wrong place at the wrong time. To this day, according to some sources, a programming error that prevents the successful performance of a program task is known, in honor of these moths, as a “bug.” See Richard E. Pattis, Karel the Robot: A Gentle Introduction to the Art of Programming 14-15 (1981).

4. Charles Babbage, an English mathematician, is credited with having designed the first programmable computer, known as the “analytical engine.” His device was not an electronic computer and was not regarded as an important advance at the time. Lord Byron’s daughter, Ada Lovelace, was Babbage’s programmer. Von Neumann was, therefore, in some sense, the reinventor of the idea of making a computer programmable.

The idea that data and programs are different ways of looking at the same thing is key to many of the conceptual problems that arise as we adjust to Von Neumann’s insight. As a practical matter, with only a few exceptions (for example, compilers and operating systems), data and programs are treated distinctly within a computer. Among the reasons for this is that if an instruction to modify certain data inadvertently modifies program instructions that have not been properly segregated from the data, the changed instructions can cause the whole program to crash.
and from it has come the field of computer programming and the distinctive kind of authorship it entails. With each new program processed in a Von Neumann-style computer, the machine, in effect, becomes a different machine: first, perhaps, a word processor, then, a videogame, next, an analyzer of statistical data, and tomorrow, a security system for one's domicile. With Von Neumann's discovery it was possible for writings to become machines.

Much of my work in the last eight years has been devoted to understanding the intellectual property consequences of Von Neumann's insight. The U.S. Congress has chosen to protect computer programs by copyright law, treating them not only as "works of authorship," but even as "literary works." After decades of controversy about whether computer programs were patentable either as "machines" or "processes" for operating machines, the U.S. Patent and Trademark Office has begun issuing patents for computer program processes and for programmed machines. This dual protection of computer program innovations by copyrights and patents is unusual. Until the development of computer programs, the U.S. intellectual property system assumed that a work was either a machine (in which case it was patentable if it met the appropriate standards and proper procedures were followed) or a writing (in which case it was copyrightable as long as it met an originality standard), but could not be both at the same time. But with both patents and copyrights being expansively interpreted for computer programs, it now appears possible to infringe a patent by writing a copyrighted text whose content is the subject of a patent. It is fair to say that, apart from judicial decisions ruling that one can infringe a copyrighted computer program by making and selling exact copies of the whole or substantial portions of the program code, much of the case law is about computer programs confused and confusing, but that, as they say, is a story for another day.

5. This is why general purpose electronic digital computers are known as "universal machines."


9. See Benson Revisited, supra note 6; Pamela Samuelson, Reflections on The State of
I, among others, have argued that computer programs are a distinctive kind of intellectual product and that a new law should be crafted to deal with the unusual issues they present. Computer programs do not seem to me to be "works of authorship" as that term has been used in copyright law, for the only functions of traditionally copyrightable writings has been to convey information or display an appearance. Nor do computer programs or the processes they embody seem to me to be traditionally patentable processes, for they are processes for the manipulation of information, whereas traditionally patentable processes have involved the transformation of matter from one physical state to another.

Although many questions could be posed to illustrate the complexities that computer programs present for traditional intellectual property law, let me discuss one question that I find to be an interesting puzzle: Can someone "own" a computer program language that he or she has created (or should we say "authored" or "invented")? As we shall see, deciding what is the appropriate verb to use to describe this kind of creation is itself an example of the problem. And under what legal regime might languages be owned? Before delving into these issues, it is worth being clear about why there are such things as computer program languages in the first place, and why someone would bother to create a new one.

It is only possible to communicate with a computer in a language that the computer is capable of processing. Programmable digital computers can process only two symbols: a high voltage signal (often represented in print by the digit "1") and a low voltage signal (for "0"). Computer programs consist of long strings of these binary digits or


11. Pictorial, sculptural or graphic works that have a function beyond conveying information or displaying an appearance are excluded from copyright protection as "useful articles." See 17 U.S.C. § 101 (1988) (definitions of "pictorial, sculptural, and graphic works" and "useful articles"). Concerning the implications of copyright's exclusion of useful articles from protection for computer programs, see CONTU Revisited, supra note 6.

12. See Benson Revisited, supra note 6.

"bits." In the early days of programmable computers, programs were created directly in machine language using the 1s and 0s described above.

Because programmers got tired of writing out, in full, over and over again, some commonly used sequences of bits each time the functions they represented were needed, programmers began to develop some more abstract representations for these sequences, a kind of shorthand that stood for certain bit sequences. These came to be known as "assembly languages." They were the first of the higher level computer languages, but they were still very closely linked to machine language. What made assembly language a "higher level" language was that it was a more abstract expression than machine language. Because of this, programs in assembly language have to be "translated" into machine language before they can be processed by the computer hardware.14

Over time, programming languages of a higher level than assembly language were developed including FORTRAN, COBOL, PASCAL, C, Ada and many, many more. Computer programs today are generally written in one of these higher order languages and must be processed by other computer programs in order to transform them into a machine-executable form (that is, into machine language).

There are many noteworthy things about computer program languages. For one, it is a very significant intellectual feat to develop a new language with its own syntax and semantics that can be used to create sophisticated computer programs, like those used to route millions of long distance telephone calls a day and re-route them when an errant backhoe cuts a cable. Another is that these languages have been created, not for speaking as English and other "natural languages" are, but only for writing computer programs.15 The very thing about natural languages that makes wit possible—their multiple meanings and ambi-

14. Assembler programs that perform this translation, like the compiler programs that "translate" other source code instructions to machine-readable code, treat the program instructions as data to be processed. This phenomenon is another illustration of Von Neumann's insight that computers could process both data and program instructions.

15. JAY DAVID BOLTER, WRITING SPACE 9 (1991). I have a friend who nearly didn't get his Ph.D. because although he could write programs in many different computer languages, his natural language skills were restricted to English. Finally, with some coaching, he managed to acquire sufficient proficiency in another natural language to satisfy his university's Ph.D. requirements. Since he could communicate with programmers from around the world in languages that they understood on matters related to their mutual professional work, why should he have had to acquire proficiency in French?
guities—is what makes natural languages unusable by computers, for computers must be told what to do in very precise, unambiguous and complete terms.

There are several reasons why so many computer languages have been created in the past few decades. Some languages have been developed to enable development of particular classes of applications. Other languages have been developed to take advantage of certain characteristics of particular computers. The internal processing units of computers made by different manufacturers often contain significantly different instruction sets, which render them able to process certain computer language vocabularies more efficiently than others. Still other languages have been developed because programmers have become frustrated with limitations of existing languages which they sought to overcome by creating a new language. Still other languages have been developed out of a desire for more uniformity in the programming field in order to save some of the very high costs that tend to be associated with nonuniformity. The U.S. Department of Defense, for example, sponsored the development of the "Ada" language because it wanted all of its advanced military computer systems to be written in the same language so program code could more easily be revised and maintained.

Let us take as a given that a computer program, written in a particular computer language and compiled into machine language form, is protectable by copyright law. But can a new computer language itself be protected by copyright law? A copyright law traditionalist might well laugh at the very question, for it would seem so obviously answerable in the negative. The traditionalist would say that a book describing the language and its syntax would be copyrightable, but that the language itself must be considered to be among the "ideas" in the


17. This language is named after Ada Lovelace, daughter of Lord Byron, the poet. She is sometimes said to have been the first computer programmer. See supra note 4.

18. The Ada language was also designed to make it easier to structure programs in a way that would make them more extensible and portable, as well as better able to permit different programs to share data. Another aim of the Ada language project was to make programmers more fungible. If a poorly structured programming language is used in a project, it may be extremely difficult for one programmer to take over a project a colleague had begun. Such a situation is of particular concern to the U.S. Defense Department, which moves people around a lot. Ada was thus developed to facilitate reuse of programmers as well as the reuse of program code.
book that would be unprotectable by copyright law.\textsuperscript{19} And yet, in the recent \textit{Lotus Development Corp. v. Paperback Software International}\textsuperscript{20} case, Judge Keeton rejected the argument that a computer language is unprotectable by copyright.\textsuperscript{21} In the aftermath of this case, others may claim to have "authored" computer languages and seek to enforce copyrights in them.

Suppose instead someone conceived of his or her new language as an "invention" and sought to patent it. A precedent that would seem to support the idea that a computer language could be patented is the claim in Samuel Morse's telegraph patent covering use of the Morse code in connection with the telegraph apparatus that was the main subject of his patent.\textsuperscript{22} One difference between Morse code and a computer language would be that, although the latter might have been designed to be used in conjunction with machines, such a language would not be as apparatus-specific as the Morse patent claimed. Later cases seem to suggest that new systems of representing information cannot be patented.\textsuperscript{23} So far as I know, no one has yet tried to patent a computer language. It would, however, be interesting to know what the Patent and Trademark Office would say if presented with such a claim.

IBM has reportedly developed a computer language that is used within the firm and is held as an IBM trade secret. Although trade secrecy would seem a surer method than copyright or patent to protect

\textsuperscript{19} The nearest anyone had ever come to arguing in favor of copyright in a language was in \textit{Brief English Sys. v. Owen}, 48 F.2d 555 (2d Cir.), \textit{cert. denied}, 283 U.S. 858 (1931) (shorthand system described in copyrighted booklets held to be an "idea" and unprotectable by copyright law).


\textsuperscript{21} \textit{Id.} Paperback claimed that the similarities between its user interface and that of Lotus 1-2-3 were largely attributable to the fact that the Lotus commands and their arrangement were elements of a computer program language. Paperback offered testimony by a computer scientist concerning this issue, but the judge rejected this defense as a "word game." See Pamela Samuelson, \textit{Computer Programs, User Interfaces & Section 102(b) of the Copyright Act of 1976: A Critique of Lotus v. Paperback}, 55 J. LAW \& CONTEMPORARY PROBLEMS 601 (1992). However, if pagination is copyrightable expression, see West Pub. Co. v. Mead Data Central, 799 F.2d 1219 (8th Cir. 1986), one might argue a language should be as well. Authors of two recent law review articles have, however, argued that computer languages are not protectable by copyright law. See Richard H. Stern, \textit{Copyright in Computer Programming Language}, 17 Rutgers Computer \& Tech. L.J. 321 (1991); Elizabeth G. Lowry, Note, \textit{Copyright Protection For Computer Languages: Creative Incentive or Technological Threat?}, 39 Emory L.J. 1293 (1990).

\textsuperscript{22} O'Reilly v. Morse, 55 U.S. (15 How.) 62 (1853).

\textsuperscript{23} See, e.g., \textit{In re Rice}, 132 F.2d 140 (C.C.P.A. 1942) (new method of recording music on paper not patentable); \textit{In re Russell}, 48 F.2d 668 (2d Cir. 1926) (new method of organizing information on tariffs not patentable); Boggs v. Robertson, 13 U.S.P.Q. 214 (D.C. Cir. 1931) (map projection system unpatentable).
a language, it doesn't so much give its developer a property right in the language as it protects against use of improper means to discover the secret or revealing the secret in breach of a confidence. Still, some firms may find it advantageous to program in a trade secret language, for it will make it much more difficult for anyone who tried to "reverse engineer" a program written in such a language to discern what the code says so that a competitive program could be developed. For the sake of completeness, it is worth mentioning that the U.S. Department of Defense was for a while trying to assert legal control over the Ada programming language through its registration of the term "Ada" as a trademark.

On occasion, people in the computing field have asserted that there is a need for intellectual property protection for computer languages in order to create incentives for developing them. One response to this might be that many very good languages have, in fact, been developed in the absence of legal protection. Also, it is often the case that someone who goes to the trouble of developing a new language will also develop programs utilizing the language through which investment costs can be recouped. And, by virtue of being the language's developer, that person or firm will be well-situated to do lucrative consulting work about how the language might best be used for particular projects.

But perhaps we need to learn to be more receptive to pleas for protection from those whose innovations come from computer-related fields. The computer language issue is only one small instance of the many new demands for protection which the legal system is facing now and will face in the future. If we reject out of hand all new claims for protection simply because they can't readily be fit into the legal categories with which we are used to working, we may be depriving significant innovators of the opportunity to thrive in their businesses. On the

24. There is some controversy about whether a trade secret is the "property" of the one who discovers and commercially exploits it. See, e.g., Pamela Samuelson, Information As Property: Do Ruckelshaus and Carpenter Signal a Changing Direction in Intellectual Property Law?, 38 Cath. U. L. Rev. 365 (1988). The view expressed in the text is the traditional view which I favor.


26. One of Borland International's most successful products is Turbo Pascal, a program that can be used as a tool for developing programs in Pascal. Borland did not, however, develop the Pascal language. It was created by a Swiss computer science professor. Another project from which a language developer might make money is writing compiler programs to translate source code instructions in that language into machine language.
other hand, if we become too expansive in redrawing the bounds of the intellectual property system each time a creator seeks legal protection for some innovation that doesn’t quite fit in the old system, we may come to regret this as well. Between this Scylla and Charybdis lie the intellectual property problems we will be confronting for the foreseeable future.

II. ELECTRONIC BOOKS AND HYPERTEXTS

Computers have not only brought about the new form of authorship of computer programs and new languages in which to write them; they have also begun to bring about changes in how more conventional written texts, such as books, are created, used and perceived.

There is, for example, a growing interest in making books available in electronic form. One reason it may be desirable to have text in electronic form is that in such form one can readily do automated full text searches for specific terms. Anyone with a personal computer and some electronic text to process can do quickly and easily what in the olden days was a task of significant, and perhaps even monumental, effort. Preparing a concordance, for example, of words found in the works of Shakespeare, used to be considered a major intellectual achievement. A concordance could easily take years of painstaking effort to create and was a resource greatly valued by scholars.

But once Shakespeare's works have been put into electronic form, it requires only the most trivial effort to find in seconds not only all places where the playwright used the word "heart," but all places where he used the word "blood" within ten words of "heart," something that even a great concordance author could not provide as a service. One consequence, then, of the computerization of texts has been to change our perceptions about at least one kind of printed book. Concordances are simply not as special now as they were before. Anyone who decided nowadays to devote years of his or her life to preparing such a concordance might be considered a sad throwback to yesteryear.

27. Some are even planning for the creation of electronic libraries. See, e.g., ROBERT E. KAHN & VINTON G. CERF, CORP. FOR NAT'L RESEARCH INITIATIVES, THE DIGITAL LIBRARY PROJECT The World of Knowbots (Draft 1988). Such libraries will themselves be one great book.

28. Although preparing a concordance might require arduous effort and precision, the compiler of the concordance is not considered an "author" in the same sense as the person who wrote the material being concorded. The recent U.S. Supreme Court decision in Feist Publications, Inc. v. Rural Telephone Serv., 111 S. Ct. 1282 (1991), ruled that industrious compilation alone does not give rise to authorship protectable by copyright law, although creativity in the selection and arrangement of materials may.
Printed books do, of course, often have features that aid the kinds of searches for particularized information that people now tend to do with computer data bases. Tables of contents, indexes and chapter or section titles are among the tools available to readers to help them find specific information in printed works. In their day, each of these (and even page numbers) was a major innovation in the print world, something it is easy to forget because we now take them for granted. But as we know from experience using these features, they don't always help as much as we would like them to. Indexes at the back of a book, for example, sometimes don't contain entries for the information we are seeking, even though the information is, in fact, in the text. By contrast with a selective print index, the automated full-text search capabilities of an electronic text will find all of the places where the search term is mentioned in the text, and not just those the book's editor decided to put in the printed index.

To be fair, it is worth noting that electronic texts aren't perfect either. Anyone who has ever used a computer data base with full text search capabilities knows how limited they can be. One problem is that they often retrieve too many items. One of the first lessons law students receive in Lexis training is that search terms like "breach of contract" are not worth trying because you could spend the rest of your life reading all the cases such a phrase would retrieve. Even seemingly more limited searches, for all instances of the word "buffalo," for example, will still pick up entries not only on the hairy beasts that populate Yellowstone National Park, but also ones about a city in New York state, about nickels incised with the beast's visage, and maybe even about a band that was really something in the 1970s. Someone who wants information about the hairy beast may have no interest in these latter items, yet he or she may have no choice but to spend precious time sorting through all items retrieved on an automated search for "buffalo" to find the passages pertinent to his or her inquiry.

Another limitation of electronic texts with automated full text

30. It may, of course, be a considerable intellectual achievement, when there are too many references to a particular item to list them all, to select which of them are most important to have in the index, and how to categorize those that are included so that they are optimally usable. No one, however, gets tenure for preparing a good index.
31. This is why many electronic document systems have built in a "but not" function. Much intellectual effort must go into designing an electronic document search system that will permit precise inquiries to be made by users. What aspects of this intellectual effort—adding "but not" and related search limitation functions, for example—is an act of authorship?
search capabilities is their inability to retrieve relevant or closely related items which use variant wording to refer to the search item. "Buffalo" as a search term, for example, will not retrieve entries wherein the beast is referred to as a "bison." Work is on-going in new fields of research, such as computational linguistics, to aid in the development of electronic search tools that will solve this problem. These tools will give users what they really want, and not just what they have asked for. This goal will undoubtedly be easier to achieve in the realm of scientific and technical literature because of the more restricted vocabularies these domains typically utilize. The more literary the works, the more complicated a task it will be to compute linguistic equivalents. And yet computational linguistics researchers aim to develop tools for all the searches that users of electronic libraries might want to make, not just some in particular domains.

What people really want from electronic books is for them to behave as much like printed books as possible, that is, to be usable in the same ways that printed books are usable, but also to have additional functionalities that are possible when the text is in electronic form. The quest for these more usable kinds of electronic books has led to the development of a new category of work—indeed, a whole field of research and development efforts—known as "hypertext." Hypertext is an electronic version of text that generally consists of a set of "components" of text (and sometimes associated graphics or other material) and a set of "links" connecting different parts of the text to each other. The hypertext is made usable/readable by a hypertext system program. The hypertext system supplies both the user interface with which the user/reader will interact when using the hypertext and a set of underlying functionalities, such as those that permit the user/reader to navigate the hypertext links to get from one part of the hypertext to another. Hypertext is sometimes created by

32. Carnegie-Mellon University has a Department of Computational Linguistics which is doing work on this kind of problem. See David A. Evans, Concept Management in Text via Natural-Language Processing: The CLARIT Approach, WORKING NOTES OF THE 1990 AAAI SYMPOSIUM ON TEXT-BASED INTELLIGENT SYS. 27 (1990). One way to approach resolving the problem of retrieving text that is responsive to the search request but irrelevant to the requestor's real needs is by creating electronic tags for them, so that buffalo could have an "animal" tag in one context, a "city" tag in another, and a "rock band" tag elsewhere.

33. See, e.g., THEODORE H. NELSON, LITERARY MACHINES (1974). Nelson coined the term "hypertext" and is said by some to have "fathered" the idea of hypertext. See also Hypertext '91 Proc (1991) for a compendium of recent research and development projects in hypertext.

converting a printed text into electronic form; sometimes it is created directly in electronic form; and sometimes it is created by a combination of these two methods.

To illustrate some differences between conventional printed texts and hypertexts and the process of authoring them, consider what might be involved in converting a printed book into hypertext. If the book is copyrighted, and the person who intends to convert it to hypertext is not the owner of the copyright, the converter will certainly need to get the copyright owner's permission to make the desired electronic version of the book.\(^{35}\)

If the converter is the author and the owner of the copyright, he or she is likely to discover with considerable dismay that it is not as easy a task as one might think to create a usable electronic version of the text, even when one intends for it to have exactly the same content as the printed version.\(^{36}\) Some significant additional "authoring" (or other) skills will be required to make a successful conversion of the printed text to hypertext. The converter will need to create surrogates for the kinds of usability aids readers are used to from books, but which tend to be lost when a work is converted from printed to electronic form. Page numbers, for example, can be included in the electronic version of a text, but if they are, they will have to be artificially added to the electronic text, for pages are not meaningful electronic units. Because readers/users may want to have pages as precise reference points, sometimes it will be worth taking the extra trouble to add page information to an electronic text.\(^{37}\) And while one can't put coffee stains and dog-ears into electronic texts to show readers which entries have re-

\(^{35}\) This is the only intellectual property question I pose in the text about which I feel certain I know the correct answer. Interestingly, one of the questions that the CONTU Commission was given to study was whether it could infringe a copyright to make an electronic copy of a copyrighted text. Although CONTU readily answered this question in the affirmative, the fact that Congress specifically asked them to address the question suggests that in the mid-1970s, not even this question was regarded as having a clear answer. \textit{See} CONTU Final Report, \textit{supra} note 1, at 39-40.


ceived the most use, it is possible to create, or let users create, electronic bookmarks.38

The easiest of printed texts to convert into hypertext are those that are highly structured in a consistent manner with textual components that are relatively concise and self-contained, having existing cross-references.39 Dictionaries and encyclopedias, with their discrete entries on particular topics and references to other entries in the same text, are particularly good candidates for hypertext. Not surprisingly, some of the first hypertexts have been works of these sorts. Each entry in these texts is a natural candidate to be a hypertext "component." The cross-references are natural "links" between components.

Yet even texts such as dictionaries and encyclopedias are not as easy to convert to a usable electronic form as one might initially think. The printed encyclopedia entry for "tiger," for example, might refer to "Africa" or "India" because that might be the tiger's natural habitat. Yet a searcher for information about tigers and their habitats may not want to read about the political history of either place. So when a hypertext link is created between a "tiger" component and an "India" component, some intellection will be required to make a precise link to that part of the "India" entry relevant to tiger lovers.

Compton's has developed an electronic version of its encyclopedia which is a true hypertext, not just an electronic version of the text.40 The Compton's hypertext encyclopedia not only allows users to search for information about "tigers," for example, but allows users to go from some basic textual information about tigers to more detailed textual information about tigers, to maps showing tiger habitats, to pictures of tigers, to videoclips of tigers in their natural habitats, and to entries on other kinds of cats. Each of these is a hypertext "component" and the ability to go to one from the other is made possible by the hypertext designer's creation of "links" between these components and by the hypertext system's method for allowing users to traverse the links.

The Compton's hypertext encyclopedia is very usable because its components and links are well-designed, as is its user interface and navigation system. (Its content is the same as the printed encyclopedia,

39. See GLUSHKO, supra note 34.
40. Id. at ch. 1. Compton's is owned by the Encyclopedia Britannica, so it is they who deserve the credit for this experiment.
but with some additions, most notably the videoclips.) The electronic Compton's may even be easier to use than the printed encyclopedia because one doesn't have to skim over material in which one is uninterested and one doesn't have to use one's fingers as place holders when one wants to go from one part of the encyclopedia to another. One can also make an electronic bookmark that will allow one to retrace one's steps.

Which aspects of the extra intellectual labor that has gone into the creation of the hypertext version of this encyclopedia are instances of authorship? If another firm decided to create a hypertext encyclopedia, and planned to write its own text entries and obtain licenses for different photographs for inclusion in its electronic encyclopedia, which of the features of the Compton's electronic encyclopedia could it lawfully copy? If Compton's has eight basic search functions and uses certain icons, such as a camera to signal the ability to link to a photograph or a movie clipboard to signal videoclips, could the later firm use the same or similar devices?

To protect its considerable investment in developing its multimedia encyclopedia, Compton's might well want to protect some of the usability features of its hypertext encyclopedia from imitation by competitors. To do so, Compton's might assert that these features are part of the "expression" its authors had contributed to the copyrighted work. The usability aids in printed books, such as tables of contents, chapter headings, section titles, indexes and page numbers, have become conventions and aren't protectable by intellectual property law. Perhaps there is a need to allow new conventions to arise in which all hypertext authors can share as well. Proprietary rights claims, however, may interfere with adoption of new conventions for hypertexts.

One of the things that makes systems like the Compton's multimedia encyclopedia a challenge to design is that electronic text and images are often stored in different places in different ways. Text in electronic form is stored as a sequence of character strings representing the alphabetic representation of the words. The letter "a," for example, might be stored as "00000001," and "b" as "00000010," etc. Images are often stored as sets of pixels (picture elements) that define a mosaic on a bit map of the entire computer screen. Each pixel is represented by anywhere from one bit (for black and white) to twenty-four bits (for photographic quality color). This means that images typically take up a lot more room in a computer's memory than text does, and that, in
Someone who designs a hypertext system may be able to speed up delivery of graphic images from the computer's memory to a display screen by using highly efficient graphics compression and decompression algorithms. Would these algorithms be part of the copyrightable "expression" of the hypertext product? Does it matter whether there are other less efficient algorithms available to be used? Does it matter if the algorithms are patented or patentable?

One other intellectual property issue worth noting about the Compton's electronic encyclopedia is its built-in "fair use" monitor. A user can print five textual entries from the encyclopedia during any one session, but that's all. While a user can make more copies by ending one session and immediately beginning another, the limitation on per session copying is a subtle reminder that the publisher has a preference that users show some restraint in their copying conduct. And it may put most users to just enough trouble to achieve its objective.

Another ambitious hypertext project is the Perseus Project at Harvard University. It has created a CD-ROM optical disk containing electronic versions of basic materials on the classical Greek world. Perseus will ultimately contain about 100 megabytes of textual information (that's about 50,000 pages in printed text) and 10,000 images. In addition to texts of Greek authors, it has a 50,000 word dictionary, maps and a number of other reference materials. Students taking courses on Greek civilization and art are expected to use this disk in their courses, but it is also expected to be a standard reference tool of scholars.

Because there is a reasonable degree of consensus among classics scholars about what the basic texts of the classical Greek world are (few new ones have been added for millennia) and there are fewer copyright clearances to obtain (many texts are in the public domain, although if one wants to use a more recently edited or translated ver-

---

41. The adage that a "picture is worth a thousand words" is more than borne out in electronic media. Id. at ch. 5. Bit map storage of images also makes printing the image more complicated than printing text.
42. Apple Computer, Inc. has, for example, a patent on a graphics compression algorithm. Patent No. 4,622,545.
43. GLUSHKO, supra note 34, ch. 1.
sion of a text, copyright may still exist as to that edition or translation), this mini-electronic library project is a good experiment to extend the scope of hypertext to a multidocument system. Like the Compton's electronic encyclopedia, Perseus is being designed to be far more than an electronic text. Its hypertext capabilities will permit a user, after seeing what Socrates (as reported by Plato) had to say about Sparta, to explore other sources to learn more about Sparta or what other Athenians thought about it. Or if a more recent commentator's analysis of a Platonic dialogue is on the Perseus disk, a user will be able to link directly to the dialogue to see if he or she agrees with the commentator's interpretation.

Professor Crane, who has been involved in developing Perseus, has made some interesting observations about the need to blend the old with the new in a hypertext like Perseus:

Hypertext allows academics to structure and manipulate their ideas in a radically new way, but it should also reinforce traditional scholarly activity. Those designing hypertext systems that are intended for the general academic market must be careful to support not only new possibilities, but those functions with which academics are already familiar. Crane notes that until and unless scholarly contributions to Perseus or similar systems "count" as serious contributions to scholarship for purposes of tenure, promotion and the like, it is unlikely that Perseus will achieve its full potential as a resource on the classical Greek world. Crane himself is trying to "invent" new ways to give suitable recognition for discoveries made through use and contributions to Perseus, so the best scholars in his field will use Perseus not only for teaching, but as a focus of scholarship as well.

In the world of print the only way for a classics scholar to achieve recognition for a discovery made by close analysis of ancient Greek texts is to publish a book or article about it. Suppose, instead, it were possible to add a module to Perseus that would link together the texts from which the discovery was made, and count that as a publication. Crane thinks that adding new material to Perseus could become a new form of publication, and a much sought after honor.

Now for some intellectual property questions that the Perseus project raises. No one questions that the author of an article or book, even when the work heavily references other works, has created a work of

45. Crane, supra note 44. Id. See also William Gardner, The Electronic Archive: Scientific Publishing for the 1990s, 1 PSYCHOL. SCIENCE 333 (1990).
authorship that does not infringe its sources' copyrights. If instead one writes a module of electronic text that consists of a set of links among texts, is that a work of authorship? Is it copyrightable, or is it only an "idea" or "discovery" that under section 102(b) of the copyright statute is unprotectable by copyright?

Is the module a derivative work of the underlying works of authorship for which, if a copyright is still in effect, permission would need to be obtained to create it? Or would permission only be needed to add the module to the Perseus disk? Interestingly, copyright's answer to this question may be quite different than the scholar's answer. Copyright would be concerned to find out how much "expression" from the first work could be found in the second. Scholars, however, would be likely to make permission contingent on whether the module had been recognized as containing a good discovery, that is, based on the merits of its "ideas."

Suppose some of the individual users of the disks (who bought it as a text for their classes) decide to add an unauthorized module to their disks without permission. Would that be "fair use"? If a particular teacher required students to add his or her module to their disks, even though senior people in the field had not yet anointed it as a worthy "add-in" to Perseus, could he or they be penalized?

Yet another hypertext experiment that is worth mentioning here is one designed for writing hypertext fiction. Some English professors have written a hypertext system program (named Storyspace) which is intended to permit authors of fiction to create hypertext stories. Its


47. Section 102(b) states in full: "In no case does copyright protection for an original work of authorship extend to any idea, procedure, process, system, method of operation, concept, principle, or discovery, regardless of the form in which it is described, explained, illustrated, or embodied in such work." 17 U.S.C. § 102(b) (1988). The Supreme Court's recent decision in Feist Publications, Inc. v. Rural Telephone Serv., Inc., 111 S. Ct. 1282 (1991) ruled that facts were unprotectable by copyright law because they were "discoveries" under § 102(b), as were uncreative compilations of facts, such as white pages listings.

48. One decision suggesting that such a module might not be an infringing derivative work is Lewis Galoob Toys, Inc. v. Nintendo of America, Inc., 20 U.S.P.Q.2d 1662 (N.D. Cal. 1991) (although defendant's program interacted with and changed some aspects of the play of plaintiff's program, the court found no infringement on a derivative work theory). The developers of the Perseus disk have thus far actively encouraged users of Perseus to create their own modules.

49. Jay David Bolter & Michael Joyce, Hypertext and Creative Writing, HYPERTEXT '87 PAPERS 41 (1987). Interestingly, Storyspace is now being used as a hypertext system for a project
authors intend that hypertext stories created through use of this program can change with every reader and every reading, for they need not be read in the same linear fashion that has characterized printed text. One of its authors has written a hypertext story using Storyspace to illustrate its potential. Another English professor has used Storyspace to make a hypertext version of Jorge Luis Borges' story "Garden of the Forking Paths."

Such hypertext stories bear some resemblance to the adventure videogames our children play on their Nintendo computers. These games start a player in one situation. By reacting to various events (such as fireballs being hurled at our hero), the adventure will unfold in this way or that. A player will either be quick enough to grab the treasure and use it to buy the appropriate weapon, or he will die an ignominious death, and so on. Storyspace's authors insist that although the texts of videogames have thus far been somewhat simple-minded, the methods of game presentation are not, and can be put to use for serious fiction.

Storyspace has two modes, one designed to be used by the hypertext fiction author, and one designed to be used by the reader of the hypertext story. The author can use Storyspace to write a series of "episodes" for inclusion in the story using a structural editor which gives the author a graphic or diagrammatic view of the hypertext as it is being created. Each of the episodes is a hypertext component. The author can choose to make explicit decision points to serve as links between episodes (for example, "Do you want to hear more about what happened to my mother in the summer of '43?" If yes, the link is made to more details; if no, a link can be created to a different episode in the story), and whether to make links two-way or only one-way, and whether some links should be of different types. Or the hypertext author can create more implicit links, so that the reader can indicate an interest in learning more about something by highlighting it. By exhibiting this interest, the reader may affect the range of available choices when reading subsequent episodes. The author of the hypertext can see and manipulate the whole pattern of connections between and among

in the state of Michigan to put judicial "bench books" into electronic form.

50. Id.
51. Bolter, supra note 15, at 123-139. Joyce's hypertext fiction is called Afternoon and is published by Eastgate Systems, which also publishes Storyspace.
53. Id.
54. Id. at 43-45.
components. A reader of a hypertext story with Storyspace will generally be unable to see any of this high-level structure. All the reader will typically see is the contents of each episode as he or she responds to questions posed at the decision points or otherwise seeks links to other parts of the text.

While one may agree or disagree with Bolter and Joyce's characterization of this system as a revolutionary one, in the experimental tradition of Dada, Jean Arps, Tristan Tzara and the Tel Quel Group, there is no question but that writing a series of episodes and decision points for linking episodes together is yet another new kind of authorship made possible by advances in computer technologies. Even converting a Borges story from print to hypertext requires thinking about its text in a radically different way, for one must plan how its text can be carved up into a set of components, how these components should be linked to one another (which links to allow, which not to), and what (if anything) to say at decision points.

Although Storyspace readers can make choices along the way that will significantly change the "story," there is still an "author" in control of the network of episodes that make up the story. Because of this, a challenge to a hypertext author's copyright in the story based on the ability of the reader to vary its order seems unlikely to be any more successful than were previous challenges to the validity of videogame copyrights by competitive copyists who argued that the player's role in determining the action of the game made the players, rather than the programmer, "the author." On the other hand, if a reader constructed a particularly insightful path through the episodes of a hypertext story—one which became "the definitive" reading of that text (as if there was such a thing)—I am not willing to dismiss out of hand the idea that there might be some original authorship by the reader that might deserve copyright protection as well, even though the act of reading itself has generally not been considered by the law an act of authorship.

55. Id.
56. See, e.g., Stern Electronics, Inc. v. Kaufman, 669 F.2d 852 (2d Cir. 1982). On the other hand, the Second Circuit's dismissal of the challenge to the video game copyright was largely based on the constancy during play of particular graphic and auditory features of the game. Id. at 856.
57. The idea that readers might participate in the construction of a text through their interpretation of it, and that this might be an act of authorship, has gained some currency in modern literary theory. Copyright law, however, has been largely insulated from this trend. See Peter Jaszi, Toward a Theory of Copyright: Metamorphoses of "Authorship," 1991 DUKE L.J. 455.
CONCLUSION

It is with some dismay that I end this essay without having discussed a great many other examples of new kinds of authorship made possible by computing. Let me mention just a few of them in closing. A colleague of mine at the University of Pittsburgh School of Law has created an expert system program for analyzing trade secret cases, and is now working on a “courseware” program to train students about trade secret law.58 Others I know are constructing digital library systems.59 Still others are creating new experiential worlds, which go by the name virtual realities,60 and neural network programs whose content changes in response to the learning they have been programmed to do from examples given to them.61 Closer to the concerns of conventional authors are some computerized authoring aids that some regard as able to make intellectual contributions to an author’s writing.62 There are even some automated text generators that have produced some short works of some appeal.63 These and other works are extending the concept of authorship in ways it is difficult sometimes to comprehend. The intellectual property questions these new works will pose will also require some extension of old concepts. Though there is much more to say, this essay is at least a start on exploring how computers are changing our concept of authorship and what intellectual property consequences may flow from that change.

hypertext, users might create a “bookmark” trail through the hypertext that could itself be a separately stored work which its creator might charge others for use. See NELSON, supra note 33.


59. See supra note 27. See also Gardner, supra note 45.


