Transcript of

Intellectual Property in Computing: (How) Should Software Be Protected? An Industry Perspective

Frank Ingari, Vice President, Lotus Development Corp.
Mitchell Kapor, CEO, On Technology
John Landry, CEO, Agility Systems
Tom Lemberg, Vice President & General Counsel, Lotus Development Corp.
Randall Davis, Professor of EECS, MIT (Moderator)

Massachusetts Institute of Technology
Room 26-100
7:30–9:30 PM, October 30, 1990

Organizers:
Michael Ernst, mernst@lcs.mit.edu
Randall Davis, davis@ai.mit.edu

Randall Davis: Good evening. I'm Randy Davis and I'd like to welcome you to the first panel on intellectual property protection of software. I refer to it as the first because we view this as the opening round of an important and interesting set of discussions that we think may help to shape the future of intellectual property and the software industry.

Let me begin by acknowledging several people and organizations that were involved in helping to make tonight's event possible. First and foremost is Mike Ernst, a graduate student at MIT and member of MIT's League for Programming Freedom. Last spring Mike originated the idea of having this panel and has put in a considerable amount of time in the months since in making it happen. Second, we have received significant support in a variety of forms from several sponsoring organizations including MIT's Artificial Intelligence Lab, the Lab for Computer Science, the Center for Coordination Science, as well as the Boston Computer Society and the Massachusetts Computer Software Council. Third, of course, are our panelists, who have graciously given their time to tonight's discussion.

Speaking of the panel, while we intend a freewheeling discussion, we did indeed select the panelists with some idea of the perspective that they'd be bringing to the table. By way of introduction, let me sketch out the role I've asked each of them to play.

Frank Ingari of Lotus Development, who is currently in charge of the Emerging Markets Business Group, was until recently head of Lotus's PC Spreadsheet Division, and in that role had considerable first-hand experience with a variety of intellectual property questions. It would hardly be shocking to you, I'd suspect, if he were to argue for the broader protection position.

Mitch Kapor, chairman, CEO, and founder of On Technology, and founder of Lotus Development, has extensive entrepreneurial experience in the software industry and has testified before Congress about protection for software. As many of you know, Mitch has also recently become concerned about and involved in issues of civil liberties surrounding computers and information.

John Landry, chairman, CEO, and co-founder of Agility Systems and previously Executive Vice-President of Development of Cullinet Software, has been involved in the creation and development of numerous software companies. Both of these two gentlemen have considerable first-hand experience in building software companies and have been articulate spokesmen for the narrower protection point of view.

Tom Lemberg, who is chief council of the large software company based in Cambridge, is also an expert on international law and a variety of attitudes toward intellectual property around the world. His role here tonight is to contribute a perspective on the international consequences of U.S. legal and business practices. His company affiliation is incidental and unfortunate because, of course, we're aware of the appearance of imbalance in having two people from the same organization on the panel. It was of course not our intent to create the imbalance; instead we're simply looking for the most articulate spokesmen on a wide set of issues, and I think we've accomplished that.

You may also have noticed that we're one panel member short this evening. Lest we be accused of false advertising, let me note that we found out yesterday that John Warnock from Adobe Systems would not be able to be here. So unless he's on a last minute flight, I think we're going to have to do without him this evening.

It's indicative of how much interest there is in this subject that since the publicity on this panel started, Mike and I have gotten a lot of suggestions about the makeup of the panel. Some of the suggestions have second-guessed our selections; some questioned our objectivity; and some questioned our ancestry. It's been suggested that we're capitalist tools for including only representatives of big software companies, and it's been suggested that we've erred by omitting big software companies, like DEC and IBM. So it's clearly a no-win position.

All of that makes clear that there's a wide variety of perspectives out there, many more than we could conceivably have fit on one panel—which is where you come in. We've structured this as a short set of statements from the panel in order to ensure that there is significant time for questions and position statements from the floor in a way of getting aired as many issues and views as we can. Be assured that your opinions are welcome and will be heard.

As a way of setting some focus for this discussion, let me say a little bit about both what is and is not our intent tonight. We're not here to discuss patents and copyrights and what they do or do not cover, nor what they should or shouldn't cover. Our agenda is broader than that, and we don't want to be constrained by the existing mechanisms. We might use them as comparison points, arguing perhaps that protection as strong as patents might be a bad idea or that protecting something more than literal code is important; but the focus tonight is on what kind of protection would be appropriate, rather than on the mechanisms that we currently happen to have on hand. We're also not here to argue how to interpret previous court cases; that's the past, and we're really interested in designing the future. We're not here to speculate about where the law is or might go; we're interested in what makes sense for industry and for society.

While the panel has been described so far in terms of software, in the larger sense, we're also here tonight to grapple with the changes being brought by the growing significance of information and the consequences of a digital medium. Digital information, in particular, has, as I'm sure you know, some almost magical qualities to it.

It is, first of all, orders of magnitude easier to replicate. We now treat floppy disks as pieces of paper, yet they hold the equivalent of 1500 printed pages and require only a few seconds to copy; that's three orders of magnitude difference.

Digital information is trivial to distribute worldwide, almost immediately, over the existing infrastructure of phone lines; we need no additional distribution channels.

Digital copies are indistinguishable from the original, and that's true for software and text, as well as music, photographs, and anything else we can cast into that form. No other medium seems to have that remarkable property.

Digital information has a breadth of descriptive power that we're only beginning to explore. We use it to capture not only software and text, but music, pictures, speech, shape descriptions, and who knows what else.

Hence we have a wide variety of expensive and complex finished products that can be duplicated exactly at effectively zero cost, and distributed almost instantaneously around the planet. What other valuable commodity has ever had that set of properties? All of that upsets our existing mindset, because that mindset is based on experience with physical property. Much of it relies implicitly on the barriers around duplication, use, and distribution that exist for physical objects. Those barriers are considerably eroded by digital information. The game is changed and is going to keep moving in this new direction for quite some time to come. The changes are profound enough that the old ways of doing business and the old ways of thinking may simply not work anymore. Hence the discussion here tonight.

Recall that our focus is on the shape of the industry and remember that when you argue for one or another approach to protection, you are in the same action setting the terms for competition. No matter how the Lotus decision, for instance, might have come out, it would inevitably have set the grounds for how business is to be done in the software world. And because the issue is larger than software and encompasses much of what the digital medium brings, our focus is also on the shape of society. How do we come to grips with information as a tangible commodity?

Finally, a few comments about how to think of all this. First I suggest it's important that we think globally. We live in an international economy now and can't avoid the global implications of the decisions we'll be making. Second, I want to urge you to think in pluralities. Keep in mind the multiple constituencies, all of whom have an important stake in what we're about here. First there are the software companies, from the giants to the one-person startups. Just to give you a sense of it, in Massachusetts alone the software industry employs something on the order of 50,000 people. Three quarters of those companies have twenty-five or fewer people. Three quarters of them, roughly speaking, have $5 million or less in annual sales. So in this state we've got quite a significant collection of rather small companies and businesses. Other constituencies include: entrepreneurs, who want to start companies; researchers, who need to share ideas and techniques; academics, who need to be sensitive to the consequence of decisions about intellectual property for the mood on campuses and in laboratories; fifth, what we might call the software artisans, programmers more akin to artists and craftspersons, who happen to work in software rather than in clay or oil, but who nevertheless do it largely for the joy of creativity; and no doubt other constituencies that I've neglected to mention as well.

Finally, think about the larger context. While the panel description declared that it was time for the software industry to begin to shape its own future, remember that intellectual property has at its heart the public trust. As the Constitution says, the intent is to promote the progress of science and the useful arts. Intellectual property may be the means to that end, but in examining the machinery, let us not overlook the fundamental intent and goal. We're not here to design the future of the industry for its own good, but for the benefit of the larger society in which we sit. It's appropriate that those who understand this wondrous stuff take on that challenge, but let us remember that there is an issue of public trust here as well.

One final remark about logistics on all of this. I've asked each of the panelists to make a brief statement, after which we'll open the floor briefly to rejoinders from the panel, and we'll devote the rest of the discussion to audience participation. Given the importance of hearing from lots of sides on these issues, audience members are welcome both to address questions to the panel and to make position statements of their own on any of the issues. Both pointed questions and emphatic statements are fair game. We'll enforce only two ground rules: civility and brevity. We expect you to be at all times civil (that goes for the panel as well), and in order to give as many people as possible a chance to speak we will enforce brevity by borrowing from football the concept of the two minute clock; here it is. You'll have two minutes to state your case; please try and stay within that. As moderator I reserve the right to trim statements that run significantly beyond that. We are also, by the way, being videotaped and, I understand, audiotaped for posterity and broadcast over MIT cable TV.

With that, let me turn to the panel and ask Frank Ingari to kick off the discussion.

Frank Ingari: I'm going to try to talk specifically about intellectual property protection in software. I've been asked to play a role; and the role is the advocate of strong protection, which is something like the sacred acolytes at Chichen Itza, who were asked to take the role for a year at the end of which they get thrown into the sacred pit. That's my feeling here tonight, anyway. But I'm a brave if foolish lad, so I'm going to take the challenge heartily. I'm not speaking—I'm going to make two disclaimers: While I am very much behind the Lotus position, I am speaking tonight as an individual, not as an officer of Lotus. I say that by choice, I wasn't asked to say that. Secondly, I'm not a lawyer; I've looked at this from the standpoint of a business person, and I want to say (you can believe me or not) I've thought about these issues as seriously as I know how from the standpoint not only of a large company, but from the standpoint of small companies as well. I take my job as a trustee of the Massachusetts Software Council very seriously, and look at small and large companies all the time. So, right or wrong, I'm trying to think in that mode.

Significance of this issue: This is really a mammoth issue. Decisions and feelings and positions taken today will have, I believe, major impact on the world. That's a strange phrase for a business person to ever utter. We're usually thinking about market shares and points of margin. This is really a fundamental, economic, and quality of life issue, I believe.

Why? Well, first of all, I want to start with a statement that some people may dispute, but it's a place that's a sort of bedrock for me. I believe property protection is fundamental to the proper functioning of societies. Two or three years ago, I might have stood up here and said to the proper functioning of capitalist societies. I think right now that statement would be too narrow. You can go to almost any country in the world and people would say the management of property is important to how society functions in any form of society. If that statement is taken as true the following statement is an important one, one that people may not agree with. And that is that if property protection is fundamental to the proper functioning of societies, then intellectual property management, intellectual property protection, is a critical issue today for everyone. And there's a number of reasons why that's true. First of all, the notion that property can be only physical, material, corporeal, is an incredibly burdensome and frightening notion to me. To say that someone who can manufacture a widget has the right to property protection but that someone who writes a symphony does not seems to me to be a very troubling notion and would reduce the very creative people in society to the state of having to depend on powerful patrons who own the material world. So to me it's critical in order to recognize the higher achievements of man, and also, from a much more narrow perspective, it's important in understanding how countries, states, and individuals will prosper in the world economy to come. To put it very bluntly, if we don't have intellectual property protection, we are giving up some very, very important positioning for our state, our country, our industry in the world economy, because intellectual property is the best property we've got. We are very poorly positioned to compete in the physical world coming down the road; we are very well positioned to compete in the intellectual and spiritual domain, I believe. It is an urgent issue today in global competition in the way that companies are battling with each other, positioning with each other, and in the way that countries are positioning with each other. This is a fact. Taiwan, Singapore, Italy—there are companies there manufacturing blatant copies of works done in this country. There is significant lobbying going on by countries who have an advantage from a cost/production standpoint in the material domain to reduce intellectual property to a commodity that can be freely copied and reverse engineered. I believe that's extremely dangerous and important. That's why I think this is an issue with great significance.

Well, if intellectual property protection is important, the question is, how should it be managed to the best ends, what are the boundaries, what is the degree of intensity? I think you have to look at the goals. What do we really want to do in society in regards to intellectual property protection? First, we want to reward the creator of value for the customer, for the consumer. That's where we have to start in our explication of value. It's customer value; it's consumer value. Related to that, the idea is that you should have law and implementation of law that will continue to develop value for customers in the ongoing manner. And that says to me that you want to continually incent entrepreneurial initiative. The French economist Seay wrote that: "The entrepreneur takes resources from an area of lower productivity and moves them to an area of higher productivity." That's what defines an entrepreneur. Innovation is a specific tool, Drucker says, of the entrepreneur, in which we create new resources or improve the use of resources. To me that process of entrepreneurial innovation is essential, and what intellectual property law has to do is to enable startups to flourish on the one hand, to make sure that continual innovation is able to happen with very low barriers to entry; at the same time (and I think that this is an equivalent good) it must enable successful companies to flourish. Those two things are both important, and any application of law that only makes it possible for brand new companies to start up may sound romantically very correct but I think practically will be disastrous, because it will mean that the software industry is capped effectively at $10 million or $20 million or $30 million companies, which will shift the burden of power back to those companies which have huge scale; that is to say, the hardware manufacturers, a troubling notion for me. The third goal, I believe, is that we want to punish theft. That's a hard statement; I made it consciously. I thought about softening it up for this group; I thought about making it "prevent theft." But then I thought about that; that's how you get into fascism, when you try to prevent bad behavior. What we're really saying is that if you do this, then if everyone in society's mechanism decides it's wrong, then you ought to get punished for it. Now theft is a very strong term, and I'll come back to that term.

Well, if those are the goals, how do you figure out what to protect? I think the question that starts is, Where does customer value come from in software? These are simple words, but I think you have to start in simple principles. Software enables customers to accomplish something in a particular way, by specific means. I think that the particular way in which software enables someone to do something deserves some kind of protection without getting into how much or what the means are because I think those are very tough questions. We'd like them to be simple; they're hard, and this is new stuff. The rules for defining what is protectable and what should be protectable need to be developed; they are not developed. And there is a debate today between two schools of thought. One school of thought says let copyright law sort it out through the application of case history, which looks at all of these suits that are in process now; that is very important for building the accretion of experience that will make a really meaningful law in ten or fifteen years, which is the way the law works. You look back and you look at music, for example; it took 20 or 30 years of copyright application before the case law gave us a viable framework, which I think most musicians today would agree is largely a viable framework. People understand it, there's room for innovation, and like software, there's room for copying, because the creative musician cannot be afraid of a two-five-one progression. You must be allowed to have a two-five-one progression but he's not allowed to have a complete sequence of notes and intervals that duplicate someone else's melody. That's the domain-specific kind of information that gets developed over years of case law, when you figure out who was thieving and what drove them, and how did they behave, and what are the domain-specific concerns of that area of music versus painting versus film. So this debate between "let case law precedence sort it out," and "it's never going to work, copyright is wrong or copyright is bad, we need a new policy and we need it now," which scares a lot of people like me, more than anything because we're in a very vulnerable period in the world around the subject of intellectual property protection. New laws take a lot of time to draft, and no matter how well we draft the new law around software protection, it's going to take 20 or 30 years of case law in that domain to make it work. Whether we like that or not, that's how the law works.

So let's look at these customer value sources in software. I think these are seven that I've come up with, having been in the software business for ten years or so. There are others, I'm sure. I think that the customer derives value when using a piece of software from things as varied as the way that the software designer conceived of the problem solution from the actual function in the product; of course, the ability to add, subtract, multiply and divide, in 1-2-3, for example. A third area of value derivation is that which enables the user to achieve the desired result—the user interface. And I'll come back to this in detail. Fourth there are what I like to call conceptual data structures which the user is taught or which are illuminated to the user, provided to the user, and the user then uses. In 1-2-3, for example, there's the concept of a range to describe a set of values in a cartesian coordinate system; Excel has a similar concept called the array. Then there is the actual internal implementation. For example, the algorithms, internal data structures, code detail—the actual writing of the code. This value also comes from bundled data, that comes with the program code; things like fonts, font outlines. Also, for example, marketing information, or typesetter width tables. And then finally, value comes from usability with other hardware and software. There are many others, I'm sure. Each one of these is a topic in itself, and what I've just made by, in our community today, is that we're all fighting about "Is protection needed or is protection not needed; is it good or bad?" And we really should be working on the very hard work of detailing the policy on each of these areas. And maybe I'm wrong about what the areas are, but there's a lot of really hard work in answering this challenge. I believe the information technology community needs to guide society—the judges, the legislators, and so on—on how to provide the proper kind of protection, and the proper degree of protection, for each of those sources of customer value, so that we have the desired result; and that is to both allow successes to flourish (which is an extremely important principle, I believe, in terms of developing ongoing quality and excellence), and also, making it such that the barriers to entry are not so high that small companies cannot enter the market.

Let's talk specifically about an area that I think is subject right now to a massive amount of misinformation and disinformation. And I'm very happy to have been given my prop here when I came in, *Against User Interface Copyright*, which I think misses exactly some of the points that I would like to focus on. And this is not a setup because I had done my foils before; I promise.

What is user interface? This is obviously matter for debate and discussion, but I would say that the user interface does consist of tools such as pull-down menus, moving cursor menu bars, and so on; it consists of conventions (the MacUser Interface Style Guide; the many conventions around the use of 1-2-3, such as the moving ring cursor); it also consists of inventions, things that didn't exist before (stuff that came out of PARC—everyone has some idea of that). But most important, I believe, user interface has to do with the uses of those tools to solve a particular problem. It's not that a pull-down menu doesn't do much for me; it's when I pull it down, what's in there: What words are chosen, what functions are available to me? To me, a way to describe it is a user interface is a creative expression of an idea, of how best to enable a user to achieve something in a particular way. And the way that that happens is 1) by using UI tools like the pull-down; 2) by adopting, by flouting, or by proposing new conventions; and 3) occasionally by inventing a UI technique, something that just doesn't exist—it's unique. But I think especially, a user interface is the sum total of that which enables access to the program function by linking your usage of those UI elements (tools and conventions) and the program resource, the program data structure. It's the program function: It's the holistic merging of those things. That's not a nice tidy description of what a UI is, but I think in reality that's what it is, and, unfortunately, we're going to have to help develop a case law that deals with that soft and fungible nature of software.

So what should be protectable in UI technology? This is my position, not Lotus Development Corporation (I don't think we would differ much). Basic tools, we believe, should be open. I think that it has been very convenient to sort of think that Lotus hasn't taken that position; Lotus has never argued that the moving cursor, the menu bar, or the inverted 'L' were Lotus inventions—never argued that at all, and still don't. We are supporting the UI's, like the Mac UI, like Windows, like PM—we believe in that stuff. Tools should be open. Conventions should be voluntary. People should put them out. You pay the price in the Mac world if you don't adhere to the UI convention. True invention, I believe, should be protectable. Someone who comes up with a brand new way of doing something should have some kind of protection. I myself feel very very nervous about patenting in software. I don't know what the right levels are. But I think that at some level an inventor ought to be able to be rewarded and have some protection. But I do think that the integration of the UI element application—how you use this total sum of tools and conventions in their total effect—should be protectable. That is, how the program enables the user to accomplish something in a particular way; at some level, some way, we have to figure it out, a person has got to be rewarded for that or the structure's going to break down. And unfortunately, not very nice from a political standpoint, the work, the devil, is in the detail and in the degree; and there's a lot of hard work that's going to get required to sort these problems out.

Now, to give you an idea of what I mean by this sum total: This is the 1-2-3 release 2.01 menu system—this is the entire menu. Now people say, "Oh, you guys are suing those guys because they used the same top level menu bar." I've left out one menu here, which is graph, and that's this page. This is about 97 or 98 percent of all the program structure and all the user interface access in 1-2-3. Every single node—every single node—on these two slides is present in exactly the same place, and exactly the same detail in the two products that we sued. This is not a question of "They used a moving cursor menu bar and we don't like that." This is a question of taking all the work that was done, all the engagement with the customers, all the refinement and testing, and then taking the documentation and using that as your functional spec. If this is not within the bounds of protection, then we have a very interesting world ahead.

Final slide. As I said to someone on the way over here, "If I'm going to wear a target, it's going to be fluorescent red." Suitability of copyright. I believe personally (I'm not a lawyer) that (but I'm a musician; that's what I was doing before I came to this business, and it works for me) the concept of idea and expression, which is fundamental to copyright, to me is a very good analogy (maybe not perfect; I haven't heard a better one) for functionality implementation; for doing something in a particular way; for doing the same thing in the same way. Copyright also has a proven ability to meet industry-specific or domain-specific needs over time via the accretion of case history. There is substantial value in the current copyright law and case history to guide decision making, and finally, the application of this law can be influenced by community leaders, and I don't just mean industry. I mean academia, I mean users, customers, all the people who care. So I think that it's a viable vehicle and I am concerned at this particular moment when the world order is changing so dramatically and intellectual protection laws are about to be signed or agreed to among many nations, that if we in the States who are driving that movement suddenly lose our conviction about intellectual property protection, I know very specifically what that's going to mean to my company and to other companies. It's going to mean major losses to us from people who see a good opportunity not to come to terms with the intellectual property demands of the US and the other Western countries; we'll just go without.

Just a comment—I won't go through in detail, but there's a statement in here that I think elucidates some of the apparent difference in opinion which I think is not really germane. The comment is made in this document *Against UI Copyright*:

> Copyright on a user interface means a government-imposed monopoly on its use. In the example of the typewriter, this would mean that each manufacturer would be forced to arrange the keys in a different layout.

I think that's a very simplistic description. By my description, the letters of the alphabet would be tools. Of course those should be open. Any way you arrange them is fine. But to say that the letters of the alphabet have the same level of semantical, philosophical, contextual meaning and significance as the arrangement of words I think is clearly fallacious. The same kind of notion comes up later when the comment is made: "Join the League for Programming Freedom, users opposing software patents and interface copyrights. The League is not opposed to copyright on individual programs." I believe that as years go by, we will find that interface, broadly defined as I have defined it (which I think is a meaningful definition) will be harder and harder to separate from the content and value of software. And to be opposed on the one hand to UI copyrights, unless you're talking about copyrights on tools and basic elements, and yet to propose protection for software packages I think will turn out to be sophistry over time. Thank you.

Davis: Thank you, Frank. And next, Mitch Kapor.

Mitchell Kapor: Well, I hope you'll forgive me if my remarks are less than fully linear, but there's a lot of ground I want to cover in a short period of time, some of which comes from prepared remarks, some of which come from notes that I've generated while listening to Frank speak. So I hope that the value of the content will make up for gaps in the continuity.

I have to make a number of meta-remarks about this on the assumption that not everyone knows all of the inside moves, and it's hard to tell the players without a scorecard. On the copyright issue, you have a clear opposition on the one hand, between Lotus's public position, and the LPF, the League for Programming Freedom. But on the patent issue, you happen to happen to have a complete convergence of interests between those two groups that are opposed on the copyright issue. And it's instructive to understand why. I'll give you my opinion on this, which is that—I know in trials, lawyers are not supposed to make remarks about the various actors, but they're supposed to argue the points of the law. I'm not a lawyer, so I'm not bound by that—and sometimes, frankly, I feel a bit like Doctor Frankenstein. (Think about it.)

In order to undercut the notion that people's point of view is dictated by their political and power interests, which is what I strongly believe (and I'm going to come back to some analysis on who takes what point of view based on political and power interests) you have to have some way of looking at the whole issue that doesn't immediately get you swamped by wondering who to believe. I go back to the Constitution. The Constitution is pretty clear. It says that Congress shall have the right to grant these partial monopolies to the creators of intellectual property in order to promote progress in the useful arts and sciences. I don't think there's any disagreement, then, that intellectual property protection is an instrumentality for social good. And it's the social good of everybody that ultimately has to be the criterion, and not the interests of the particular holders of intellectual property rights that has to be taken into account. By that standard, too much protection is as bad as too little protection if it fails to promote progress in the useful arts and sciences. I'm going to use the word innovation as a shorthand substitute for promote progress. But my reading of the Constitution says that too much protection stifles innovation, and that's bad for society. So what the debate is about is how much protection is just enough. And you have a spectrum of positions from minimalist to maximalist, in the words of Pam Samuelson, who is one of the leading attorneys on this.

I'm a minimalist. I'm a minimalist fundamentally because I think you get the maximum net innovation by restricting intellectual property protection to a reasonable minimum. Today I would call the reasonable minimum in the desired sense the protection of source code and object code, the literal elements of the program, and no copyright protection for non-literal elements, user interface, or anything else. This happens not to be the state of the law today, after the Lotus ruling. So in a practical sense, in order to achieve what I consider to be the minimum stable point of law, to regain a stable point of law, temporarily it's going to be necessary for Congress to pass legislation clarifying their intent as far as copyright, which is to restrict it only to the literal expression of a program at the source code and object code level. I'm trying to lay out what my copyright position is. As far as I know, that happens to coincide with the LPF's position as to what state they would like to see things in.

My reasons for it are pretty simple. I don't think anybody's ability to innovate is hurt in that position, whereas I think positions that accrue more protection to the creators of intellectual property hurt net innovation. It's a very simple argument, and I don't believe, for instance, that Lotus's ability to compete is at all significantly impacted by the two companies that were sued. And despite the fact that Lotus's ability to compete is impacted by Borland, it's on account of the fact that it's competitive problem, and for no other reason.

Which brings me to a fundamental point, which is where I started (to close the first loop): It's necessary to understand that the positions that all of the parties take, not just the corporations, are influenced by how they perceive their self-interests. Lotus and other software companies tend to perceive their self-interest as being that of getting more protection, and they don't particularly, as far as I can tell, take the position of looking out for what is good for society. It leaves it up to other people, I suppose, like me, if they don't agree, or even if they think it is good for society—it leaves it up to other people to take up an opposing position about where the good is. But my experience in the corporate world tells me that in fact it doesn't intend to be a reasoned set of debates about what's good for society; that policy decisions are made about what's good for business. That's just reality. We can like it, we can not like it, but it's not a basis for finding a principled kind of stance to take.

So what happened in the software industry was that up to the point—what's been happening is that there's been a stubborn lurch forward periodically in the case law of the courts to extend the scope of copyright protection over the past several years, as businesses have found it in their interests to litigate to prevent other people from competing or doing things. So we started with protection just for source and object code and it's been lurching forward, and the last major lurch forward was in the Lotus vs. Paperback decision. Before that decision came down, I think in the software industry as a whole, the basic feeling was that, well, look and feel copyrights were okay, or there were sort of mixed opinions about it, but nobody in the industry was really concerned that they were a bad thing. Nobody really cared about Paperback Software. Adam Osborne is not a likeable guy, and that's the kind of analysis that people actually do; I'm telling you the truth.

The decision comes down in the Lotus vs. Paperback case fundamentally extending, creating a user interface copyright. The judge said, "This is what I'm doing, and if you don't like it, you're going to have to go to Congress about it." And within a week, the tenor of the industry changed as a result of the Lotus-Borland suit, where people in the industry began to realize that their ox could be gored, too. So it wasn't like there was a fundamental reaching back to first principles that changed the climate of opinion in industry. What changed it was a sense of, "Where is it all going to end?" And that is, in fact, the problem: that nobody, as far as I can tell, has been able to come up with a principled way of extending the scope of protection that doesn't entail bad effects of curtailing people from doing all sorts of terribly legitimate things. I'll give you a practical example. If you take Judge Keeton's ruling in Lotus vs. Paperback, if I were a company that had a copyright on a computer language, something without any user interface, like Apple has on Hypertalk—though they've chosen not to enforce that right—if I were the CEO I would go to my attorney and say, "Hey, can we use this Lotus decision to sue the bastards who went and made a compatible version of our language?" And the attorney would probably say, "Well, I don't know if it's a sure thing, but you've got a good shot at it, because we take the language the way it is here in this decision, and we can make these arguments, and we have a lot of money, and we can tie these so-and-so's up for years in court." That's the real world, folks. And a lot of my objection to this steady expansion in the scope of copyright is that's exactly what goes down, and it's a bad thing. And I don't have to argue in front of this audience why it's a bad thing. There are lots of good reasons here.

I think, though, that as far as copyrights go, Frank really does have a point. And I want to say what that is. Which is that, in my opinion, pulling back to protection for source and object code is purely a holding action, and a temporary measure until we can figure out what to do really. Because it is the case that more and more of the value in the creations is in the nonliteral elements. It's in the idea and not in the expression, and our current intellectual property system has no way of recognizing that effectively because it was based on a development with artifacts that were created on physical media like print, where you could clearly have a separation between idea and expression; and software just makes hash of that.

So we have the dilemma that, if in the long term, we do intend to provide some degree of protection to the creators of intellectual property, we're going to have to go back to first principles and think about how to do that. (Today's debate is about look and feel and the skirmishes and litigations and business tactics are completely the wrong atmosphere in which to have that kind of debate.) That debate does not happen now. But it really needs to happen over the next decade because it's going to turn into an enormous mess. The real problem is not in copyright; it's in patents. Over the past few years there have been several hundred software patents issued on all sorts things, from footnotes, to redlining, to the exclusive-or operation of two bitmaps. It's a fact; it's not a joke. The reason you haven't heard about it is it hasn't been a big lawsuit with lots of money at stake, yet. But it'll happen for sure. The patent office, under Reagan-era deregulation took the policy and said, "Hell! We'll give them all their software patents! We'll let them sort it out in the courts afterwards." So what we've got is a kind of Bhopal of software patents. There are all of these terrible patents that should never have been issued because there's prior art. They probably shouldn't have been issued because they're about algorithms and the Supreme Court, the last time they said something about this, said you can't patent an algorithm. But the patent office has been awarding them left and right. And the reason this has been happening is because no one has a clue about what to do, really. So, in taking this minimalist position, my minimalist position, also includes invalidating all software patents based on algorithms. That's just a holding action to let business go on in some semi-stable fashion for a time being until we can create a climate of opinion and discussion that can sort out a long term answer to the problem. Thank you.

Davis: Thank you, Mitch. Our next speaker is John Landry.

John Landry: I started in the software business in 1972 on what was then IBM mainframes and the first product that I built was a general ledger system in a company called McCall, McCormick and Dodge in Natick, and the way that we built that system perhaps would be illegal based on the current atmosphere. We went out and we got brochures from every major company. We looked at the way the card layouts were back then formatted, the way the reports looked, the way the system kind of looked and felt, and we said, "Gee, that's pretty interesting!" and we took a piece out of this competitor's system we took and a piece out of this competitor's system—and indeed we are not dealing at all with source code or object code, or decompilations, etc., but we used them as the basis, because we thought what they had done was very good, and we decided that that would be good for us, too, and in many cases it looked just like their system. It looked like a bunch of different systems, but we went one step further—we then added a lot of stuff that they didn't have. And we put it all together in a nice package and we sold it, and luckily we sold it very successfully. And frankly, back then, that was the way that you built software. There was no sense in going back to first principles—by God, you were going out there to compete! Let's start where these guys are and let's see what we could innovate beyond that. Look and feel was not an issue back in the period of the early '70s and even to the early '80s. It never came up. Nobody ever said that you've copied my system, except perhaps at a cocktail party, where they'd rib you, but certainly never go through a legal process.

Now, incidentally, in that ledger system that we wrote, the typical model was an extract, sort, and print type report writer back then. We decided that really what ledgers needed was something a little more powerful, particularly to do such things as budgeting and financial analysis. We went and we found a program that ran on an IBM mainframe back then, which was called the ABC Financial Planning System, and no joke, it wasn't 1-2-3, it was A-B-C. And what it was modeled on was indeed the Wilson-Jones 13 column spreadsheet. And in fact, every component of that system had to do with inputting based on using a Wilson-Jones 13 column spreadsheet—it had 13 columns, it had 99 lines; the way that you addressed things was row 1, column 4; and you had math capabilities that essentially started with equal signs and then a formula; you had text that started with quote marks, and in fact, it was a mainframe spreadsheet. Now the look and feel of the thing was hard to assess because it was going in on punchcards back then. But it raises some interesting issues. Was the cell reference structure in the parlance of the courts these days "nonobvious"? Was the linking of pages together—which incidentally it could do back then, where you could reference cells from different pages in order to do such things as consolidations—was that a substantial and unique part of the system that was subject to protection? Is the layout of items on an 80 column card, in fact, any less protectable than the way it looks on a screen? Did Bricklin and Frankston, in fact, get some ideas from this system or those like it? Did Mitch and Jonathan Sacks get some ideas, in fact, from VisiCalc? This of course goes forward and you know probably where I'm heading. In my opinion, the primary look and feel litigant in this industry is Lotus Development. And in fact, my sense would be—and I'm not a lawyer either—my sense would be, based on the current court climate, it would be found guilty of copying the look and feel of VisiCalc. But Lotus was very interesting.

It was a rich enough company to really buy itself out of the problem by just acquiring VisiCalc. And now that they dominate this market they're...

Kapor: I have to say that Judge Keeton in the Lotus-Paperback ruling did rule that Lotus had a valid copyright on 1-2-3, and that it did not infringe VisiCalc. Thank you, Judge.

Davis: Which proves that lawsuits make strange bedfellows.

Landry: Bring the judge up here!

Now that Lotus dominates the market, my sense is that they're really using their power and wealth to prevent others from starting from where they were or should be from where they were, and innovating further. More or less, the sense I get from Lotus is it's okay for them, but it's really not okay for you, and I think that's foul. These are things, incidentally, that no one in the pre-PC generation, as I said, ever thought about protecting. Protecting was for code; it was for documentation; it was for trade secrets (which have algorithmic components); it was protection against theft and stealing and things that people understand should be protected. The pioneers in the business—that is, in the pre-PC business—never looked back on protecting their look and feel. The feeling was, "We've got a head start, and when the competition catches up, we'll have the new release out and we'll croak 'em." Somehow that's gotten lost in this industry and I think that's very tragic. It's what stimulates innovation and health in the industry. What doesn't stimulate innovation is essentially lawyers arguing over whether a slash key or an inverted 'L' is essential to express the idea of a spreadsheet, which was really in the decision that Judge Keeton ruled on. He luckily concluded that it did not express the idea, but it was part of the suit.

Judges should not be concluding at this point that edit, search, and file sequences on a menu bar is a substantial, distinctive, and original element of the expression. And responding to really what Frank put up there, I have a tendency to say, "So what?" How many items would have been okay? Would 90 percent or 80 percent of the copying, if there was 20 percent that was different, would that have been all right? And the point is that in my sense, what should be going on here is that there should be innovation taking place, and the way to beat Paperback Software and Mosaic is not to sue them, but essentially to innovate your way so that you have a better product, as has been done over the years, to compete against. It should be innovation and not litigation.

I direct that at Lotus; I also direct it at Apple. I think Lotus for the most part over the last eight years has not done a good job of innovating in the spreadsheet market. They've really taken what now is a rather tired character mode interface and brought it over n platforms, and instead of spending their considerable cash reserves on innovation, a not so trivial sum has been spent suing such industry behemoths as Paperback Software and Mosaic.

I was very pleased to see the pre-announcement of Improv from Lotus. It looks very interesting, a truly interesting and innovative variant on spreadsheets. The point is if that had come out earlier, or something more innovative than 1-2-3 release 3 had come out, then really the Paperback and Mosaics would be a non-issue, because they would have no competitive value against the wonders that Lotus would then have. My sense is that people have competed with like products for centuries, and my own sense is that we should not have any more product protection for software user interfaces in this case, than we should for a pair of shoes that's like somebody else's pair of shoes. So my advice would be to stop looking back. Copyright law or not at this time, this industry has been up to now been built on the achievements of others, and I think that it should continue to be that way. We should not apply legal protection to computer software interfaces whether they be language interfaces or user interfaces. It stifles the innovation in this industry by preventing developers from capitalizing on an installed base and taking the technology forward.

In patents, God help us all. First, in my opinion, it is the playground of the large companies, the ones with all the corporate lawyers that can afford to look at all the information that is coming forth on patents, and in fact to file the patents themselves. I don't believe the patent office has a clue about what it's doing, and when I heard about the exclusive-or on the bitmap that convinced me even further. I was limited to such things as backward chaining algorithms in expert systems being patented; every expert system uses, Tech Knowledge has the patent on that. But how does one even find out about it? Getting a patent right now isn't expensive, relatively; it's $10,000, give or take a few thousand, which isn't that expensive certainly for a Microsoft or an IBM. It is somewhat expensive for a small entrepreneurial company, but that's only part of it. It's keeping abreast of the ones issued and pending that only the rich companies could really afford to do. If copyright suits don't kill off innovation, I think a few phone calls from IBM claiming infringement just might. Defending against an infringement suit would put most entrepreneurial software companies out of business, and for the most part, dealing in infringements that they never even knew that they were doing.

So, maybe I'm just sentimental, but I want it really to be how it used to be. It worked then to create what is now an incredibly dynamic industry that attracted these very people into it. Now it's attracting lawyers and court reporters. Fight it out in the market place, not the courtroom. If you lose, you deserve it. Trade secret protection and copyright of code, documentation and executables is sufficient; change the laws. No patent protection for software, no copyrights for look and feel, on languages or user interfaces. Innovation, not litigation. Thank you.

Davis: Thank you, John. And now Tom Lemberg on the international consequences.

Tom Lemberg: Hi. I'm here under a terrible disability—and I hope you indulge and forgive me—and that is that I actually am a lawyer. So I apologize.

I'm also, although I have lots of thoughts about some of the things that have been said tonight and maybe some of the discussion after we all finish our prepared statements, maybe I'll have to opportunity to talk about some of those. But I've actually been asked here to talk about international issues because they're very important to our industry and it's something that I've done a lot of work on while I've been at that little software company down the road. So I'm going to just talk for five minutes or ten minutes now about that.

I think we have to remember that software is a worldwide phenomenon. Sounds pretty obvious and it's true, but software is used and sold everyplace in the world, and to increasing degrees. More software is used and sold in the United States than in any other country, at least in the PC world, because not only are we a larger country than most, but we have a greater PC penetration. But the rest of the world is catching up. As a result, the revenues of many American software companies are increasingly internationally based. In Lotus's case, the number has risen to about 50 percent; in Microsoft's case it's 60 percent; the numbers for other companies that are sizable enough to have tried to have an international presence are all trending in that direction, likely to go higher. It's also well to note that software in America—we've been the leaders in the world, and as a result, it's probable that US companies have about 60 to 70 percent of the world's software market today, and that's because we have better products; we're more innovative.

Now, there have been a lot of words said this evening about patents, and I'm going to talk about copyrights here today, because I believe that copyrights are the main and principal protection that our industry needs. I'm not speaking on behalf of my company now, but I will say that I share many of the great concerns about patent protection that John and Mitch, and Frank, for that matter, have uttered. And it's a very different kind of system than the copyright system. It protects different things and it's not at all clear to me, even though Lotus will get its fair share of patents, that we, or the public either, are the slightest bit served by that fact.

But copyright is the only protection for software producers around the world, not only against the creation of copied products, the subject of some debate here tonight, but against outright theft, against piracy, against copying in the most blatant form, which is you take the disks and you just duplicate them as many times as you possibly can without paying for the product. Now the scope of the piracy problem worldwide is absolutely enormous and mindboggling, and I will give you a few statistics, not because they are perfect, but because they'll probably get your attention. The piracy rates that we can best guess from the data that's available in this country is probably around 40 percent, far and away the least pirated place in the world. The second best place where we have data is the UK, where the piracy rate is about 60 percent. In places like Germany, it's about 75 percent; Italy, 80 percent. In some of the countries outside of Western Europe, the rates are in the high 90's. So what this means, for example, to take Italy, is that approximately one software package was sold last year for every three personal computers. That means that, if you believe that piracy doesn't exist, that two-thirds of the Italian owners of PC's believe them to be beautiful pieces of furniture which are aesthetically pleasing and that's all they do. So there is a very serious piracy problem; it's also in our country here, but it's an amazingly large issue worldwide. Best guess is that the amount of PC piracy in Germany last year was something like a billion four. So solving this problem is very important, and the only way we can solve it is through the application of the copyright law. I'll stress here that the copyright law, if it clearly protects code, is good enough to protect against this kind of piracy. So, in a way, what I'm talking about now is much narrower than the debates about whether or not we were appropriately suing people like Paperback or Borland.

But there are very serious intellectual property issues abroad. What do we do about that? Well, what we do is, for our company, we're part of an organization called the Business Software Alliance, and we go around the world trying to change customers' behavior; try to educate them that piracy is wrong, and we bring lawsuits to enforce our rights because otherwise you can't get anyone's attention. For example, a year and a half ago we sued seven major Italian companies, one of which, Mont Edison, is the second largest company in that country; piracy rates in these places, we found, were anywhere from a low of 80 percent, to 95 to 99 percent. So we're talking about—we're not talking about someone giving a copy to his cousin; we're talking about major, major corporations, some of the largest companies in the world, for example, buying one copy and making 50 (that was one case in Mont Edison of 1-2-3, or buying one copy of dBase and making 20).

In order for us to do that we need sound copyright laws. Now, when we think of copyright laws, we think about our system in the United States, and we think about the fact that we've had a clear law for at least ten years, and it clearly covers code. That's not true in many other parts of the world, and it's absolutely essential that we have good copyright laws in every part of the world, or else we can't bring a lawsuit. We can't today protect intellectual property from theft except by technical means in places like the Soviet Union or the People's Republic of China.

It's also critical that the United States is regarded as the world leader in these issues for the simple reason that our software industry is the largest, and our government has taken a major lead in the last decade in persuading other governments to pass laws that clearly apply copyright to software. Now, that's happened in most of the developed countries of the world, although not all of them. But there are some serious issues that remain. For example, a quirk in the German law exists by which, it has been reported by the leading German copyright official, that something like 85 or 90 percent of all the programs sold in Germany are not protected by copyright. And again, we're not talking about user interfaces, we're not talking about look and feel; we're talking about code. So it is probably no secret that the reason the level of piracy in Germany is as high as it is is the fact that in the last year or two it has become to be believed in that country that there is no copyright protection for virtually all computer programs.

Another issue in the law in Italy, very unclear. Italy has a personal use exemption, says you can make a copy for your personal use. Well, gee, that seems reasonable enough. Mont Edison is arguing in the litigations that Lotus and Ashton-Tate brought against them that it is a personal use for a company to buy one and make copies for all of the employees.

There has been a major issue in the European Community which I ought to talk about for just a minute. The European Community, as many of you know, is passing lots of what they call directives—they're community-wide laws—in preparation for having a unified market in 1992. One of those has to do with copyrighting software. And there has been a major debate in the EC about whether or not to allow the reverse engineering or decompilation of software for purposes of allowing people to create programs that interoperate with other people's programs. It's a very complicated debate; it's probably reaching its end now with a compromise. And you had people in that debate ranging from one side saying that everything ought to be protected and you ought never to be able to decompile, to the side, particularly led by Fujitsu, which is that you ought to be able to decompile, to copy, to use source code for anything that you want, including the making of substitute products of any kind.

It's an interesting perspective on some of the issues that we've talked about tonight (Randy, I will sneak into those issues just a little bit), and that is, "Where is the argument for strong protection and where is the argument for weak protection coming from in the international arena?" The argument for weak protection is coming essentially from two places. It is coming from large Japanese hardware companies, with everyone believing that MITI is encouraging that strongly; because everyone believes they've had a very difficult time innovating and creating their own software that is successful, and they would like very much to have the opportunity of being able to copy (as two of them presented a paper in Tokyo a couple of years ago, for example, being able to copy everything they want about the Macintosh because it's successful). And they've had a very hard time innovating so they want to encourage very weak copyright protection because they believe that then, if you don't have much copyright protection, victory goes to the guy with the biggest marketing budget and the biggest financial muscle. And if you think Lotus is a large company, you ought to think about Fujitsu, for example—they're a lot bigger. The other supporters of weak protection typically are some of the third world countries; Brazil typically takes the lead in this, and again, they expressly would like to copy as much as they can so that they can just simply compete on the cheap with companies in the United States and in Western Europe.

So there's this very strong international dimension to all that we're doing and we just can't forget that, particularly as we've been successful so far in this country in having been the leaders in software innovation. Thank you.

Davis: Thank you very much, Tom.

Do we have any brief rejoinders from anyone on the panel before we throw it open to the audience?

Kapor: One of the supporters for minimalist protection on the reverse engineering issue that Tom was just referring to is Sun Microsystems, a $2 billion US company.

Lemberg: Actually, Sun wants to be able to clone ROM BIOS and the current provision will provide that and that's the extent of the minimalism of Sun and Compaq, and a couple of the other—it's only on the ability to reverse engineer ROM BIOSes, to be able to compete in the PC world without...

Kapor: False. It's on ability to believe in open systems and the virtues that they have for promoting innovation and the public good.

Lemberg: Where is Sun when we need them?

Kapor: Right where they should be on that issue.

Davis: Any other comments? If not, let me invite anybody from the audience—there are microphones down front; I suggest the protocol be just line up at the mikes. We'll take people alternately at the microphones in whichever order you arrive. And as I said: Feel free to either raise a question to the panel in general or somebody specific on the panel, or to make your own statement, but we will try and enforce the two minute limit.

Jerrod Spool: Hello. My name is Jerrod Spool. I run a company called User Interface Engineering; we run it out of my basement. My clients are software development companies. The majority of them are 25 people or less, and what we do for our clients is we work on usability issues and we help them design the actual way that my clients' customers accomplish their work. And I'd like to argue the point that Mitch brought up that people are not being inhibited, or that Lotus was not being inhibited, by the Paperback suit. Innovation is being inhibited. I am seeing it consistently in my clients who are pausing and wondering if they should invest money in coming up with a unique way, an innovative way, of doing something, or going with the old way (that methodology brought us VT100 and Teletype technology for about 30 years). And I think that it's a big problem. I also feel that if the Lotus-Paperback suit had gone the other way—if it had been Paperback that was suing Lotus—that the same people would be at 55 Cambridge Parkway out protesting the large giant for infringing on the rights of the small companies. And from what I've heard today, what it seems to me is that the big problem here is the cost of arbitration. And that what everyone's concern is, and what my concern is too, because the last thing I need is a lawsuit, is the cost of arbitration. And maybe what we should be focusing on is a way of arbitrating these issues in a low-cost setting such that it is possible for people to protect their rights and not have to worry about people with large pockets suing people with small pockets.

Davis: Anyone in the panel want to respond to that?

Lemberg: I think a lot of the problems people have with litigation, and I assure you we have it for all of our alleged deep pockets, is with the expense and costs of litigation, and that applies whether it's copyright or patent litigation, or any other kind you can think of, and I don't have a solution to that. But it would be great to have one.

Elliot Smith: I'd like to pick briefly on a point that Frank brought up regarding the typewriter-keyboard, the arrangement of letters. In fact, the Sholes keyboard, invented in I think 1885, was widely copied, and became the prevalent arrangement of keys within two or three years. In spite of a superior layout of keys (and the Sholes keyboard is an instrument of medieval torture as far as your fingers are concerned), in spite of a superior keyboard, the Dvorak keyboard being designed in 1932 (which is almost 60 years ago now), there's no penetration of this superior arrangement of keys because everyone has adhered to the standard for so long. Now that's a phenomenon that also applied in the telephone industry in some respect. The company or industry that enjoys the knee in the curve of technological diffusion usually benefits mightily. I would say that Lotus is an example of such a company, and that perhaps its arrangement of commands is an example of such a technological innovation. Now having said all that as a sort of preamble, there is something that is very wrong with current copyright law, which is the neglect of prior art. What does the panel have to say about something that might defeat the establishment of prior art—electronic mail. If you have people working inside of the company, communicating the company's intellectual property outside, how are you going to establish any rights to the intellectual property?

Kapor: There's no standard of originality for copyright protection; simply coming up with the expression is sufficient under copyright law. Prior art is only an issue in patents. So I'm confused.

Lemberg: The only originality is that you did it, not someone else.

Smith: Well, I'm not a lawyer. What I really wanted to get at is the whole notion that you can have people who are working on cutting edge ideas, whether they are patentable or copyrightable, and I've made that mistake before, I'm not a lawyer. And they could be communicating your company's intellectual property away before you've even had a chance to establish your legal right to it.

Lemberg: Well, we all have to sort of worry about protecting the trade secrets, but copyright actually theoretically happens the minute you scribble something on a piece of paper or whatever medium your using.

Smith: Well, yeah, but that's the crux of the issue. If the guy happens to type it onto his keyboard and ship it out to the other end of the continent and keeps a local copy and the message that he sent by electronic mail arrives at the other end of the continent moments later, which was the prior copy and who has rights to it?

Davis: Sometimes it doesn't matter because copyright doesn't have a prior art standard. Once it's captured in tangible form, it's captured in tangible form. And you are then forced thereafter to protect it [i.e., protection has been established]. Communication doesn't change that.

Smith: So both people, the sender and the recipient, could capture the concept in tangible form, bring the product to the market simultaneously and there you are.

Davis: I think you're mangling the notion of what it means to capture it in tangible form. I'm going to indulge the two minute warning here.

Smith: I'm going to leave it off. The last thing I'd like to ask is that given even the possibility of this problem, do many large companies maintain any record of their electronic mail communications?

Ingari: You don't have to do that consciously. Walk around Lotus—there's unbelievable mountains of disk drives that have never been purged. That's a serious point. Do you remember the Iran/Contra? Somehow files have a persistence, like object oriented databases.

Davis: Whether you want them to or not. Okay, let's press on.

John Ramsdell: My name is John Ramsdell from the Mitre Corporation. We all hear of the appropriate quote from the Constitution that justifies copyright and patent laws, and what I was wondering is, could you tell me what the equivalent foundation is for the intellectual protection in other countries and how that different kind of view affects how they might view software?

Lemberg: Our Constitution, as Mitch quotes accurately, about the useful arts—there's a concept, typically in European countries (remember, they don't necessarily have this in their constitutions) but there are two concepts that are sort of pair. One has to do with the useful art or innovation argument. The other (and it's probably more important in European countries) is the rights of the author or the creator, and they call it sometimes moral rights, and its the notion of "how dare you copy my work," more than "we want to promote innovation;" it's less of a utilitarian test than it's just like one of the Rights of Man. In France, actually it is, it's one of the declarations of the Rights of Man from 1792 (I believe this is right) that included the right of the author.

Ramsdell: So that would argue that they would go for stricter protection.

Lemberg: It could mean that they would go for stricter protection. Their arguments about artists' moral rights, you know, that if you bought a painting you couldn't deface it, that you'd violate the moral rights of the author. So there's a lot of debate that goes on, especially in Europe, over that.

Ramsdell: Thank you.

AM4: My question is primarily directed toward the two people here that work for Lotus. As I understand it, a primary component of the claim that user interfaces encourage extension and development of the industry, that user interface is encouraging innovation. A primary component of that is the claim that without the protection, people won't be inventing new user interfaces and new clever ideas because they won't get any money for it because somebody will quickly code up the same thing, presumably, and be able to do it faster because they are a bigger company, or something. So for a small developer, user interface copyright is very important. Is that an accurate assessment?

Ingari: Yeah, it's close. . .

AM4: My question is that Lotus was founded—that 1-2-3 was starting to be released in the early 1980's. In the early 1980's there was absolutely no case law precedent of user interface copyright or any hint from the legal community that such a thing could exist. When Lotus invented its marvelous invention of 1-2-3, it did this without expecting any user interface [copyright], and there's every reason to believe that they only thought of this after they noticed that Apple had thought of this, and they said, "Aha! Maybe we can get money off of this."

My question is if this innovation is so essential to funding and it costs so much to invent user interfaces, my question is how much money does the research branch of Lotus—the people who design user interfaces and write them—what percentage is spent on designing user interfaces as opposed to writing code? And the other question is...

Ingari: Well, let me answer this question while you're thinking of the next one. The point that I made earlier and I believe it strongly, and it's a reason that I believe that protection of the literal code is just not good enough going forward (and I think Mitch and I agreed with that from different perspectives in terms of how we saw the problem); one of the reasons that I don't think that it's good enough is that I think that as we move forward into a new environment with different sets of expectation by the developer in terms of object stores, object databases, standardized graphic user interfaces, and so on, that the distinction between interface design and development and internals design and development is going to get fuzzier and fuzzier and fuzzier. And that in fact, even coming up with a number for a percentage of time people spend designing interfaces as opposed to the time people spend designing software as though those are separate things, or that internals are somehow separate from UI's because the UI is visible to the user—I think these are going to become more and more difficult and byzantine to support. So the answer, in a sense, and this may seem extreme, is that in fact, in many projects at Lotus, and I would count them among the best, each developer feels connected to the UI and that in fact, the work is fairly essential and fairly integral. So I don't know what the percentage is, but it would be a significant one—more than 25 or 30 percent of the work is involved with UI, and we certainly don't have groups of like, "You're the invisible programmers, and you're the visible programmers." The program is integral.

AM4: The distinction I'm drawing is not one between whether the user sees it or not. The distinction I'm drawing is whether it's written in a programming language or not. Menus are not programming languages and it's superfluous to pretend that they are.

My question really is why did Lotus invest this money on developing software when they're now just absolutely terrified that nobody will develop software without user interface copyright.

Lemberg: I think there are some premises that are not quite right in the question. First of all—and in some of the remarks—there's a sense that Judge Keeton somehow changed the world. Absolutely not true. Frank said this earlier: It takes time for cases to come through. There have been very few software copyright cases. Very few have been decided. There were no cases that even raised user interface questions until a case involving Broderbund that was decided four or five years ago [Broderbund Software Inc. vs. Unison World Inc., 648 F. Supp. 1127 (N.D. Cal. 1986)], because PC software was a fairly new phenomenon and it just takes time for people to put these things out and be litigated. If you read the law, there's no reason to think that it only applies to code and not to the things that are on the screen and visible to the user. And Mitch used the phrase "literal expression of the program" and if you talk about the program as code, of course a menu isn't literal expression. But it sure is literal in the sense of a literary work that people see. So, in 1983 when 1-2-3 was released, there was virtually no case law that was meaningful on this subject. Now, why didn't someone just clone it then? I don't know why. There was a small group of people who might have, in terms of really big companies that were there, big hardware companies, who were—I don't know why they didn't; we'd have to ask them. But the fact is that if the Paperback case had come out the other way, then it would be "open sesame" and perfectly legal, and presumably perfectly moral, for Lotus as a large software company, or for some even larger entity, whether it's a hardware company or not, to take that neat new product that hit the street tomorrow and reverse engineer it in three months and use our marketing muscle to win the day. And presumably, that would happen a lot. And that's why I believe as Frank does that copyright protection for the interface is essential to protect innovation. It's not just innovation by the large companies, it's innovation by anybody.

Landry: I think that's an extremely arrogant position for Lotus to take. There is no difference in PC software than there was in minicomputer software or mainframe software. The idea of somebody writing an accounts payable system, and somebody else copying the system in terms of its user interface existed well before any PC software came out. The fact of the matter is we're talking about a product that for the most part only has one screen in it. And I can't believe that we're going to get all hung up that this is such a great masterpiece of technology that we're going to sue everybody in order to do that. That is an extraordinarily arrogant position.

Lemberg: Frank put up a command structure of 286 commands in a very particular sequence. There are a million ways, at least, to do the same functionality that are in that product and to do them radically differently.

Kapor: You know, if I believed that Lotus or any other large software company would stop the litigation with pure clones, I wouldn't be so upset about it. When I went before Congress in March, I said that—literally! I had a conference with Lotus's attorneys (we talked about what I was going to say because there is still ongoing litigation) and I was balancing out the moral dilemma of "How do I tell Congress what I really think while attempting to preserve my other relationships?" and what I said in that testimony was, "Look, if the copyright law was extended just so far as to prevent people from doing clones, I really wouldn't care, because I don't think that that's the big issue. But my belief is that there's no way to make it stop there." Well, that was March; that was before the Paperback decision came out. The Paperback decision came out, and what do you know—the next week, there are Lotus and Borland going at each other, and it's not a clone case. So I think you've got to look at not the question of "Where has it been?" but "Where is it going?" Let's assume for a moment that the people of Lotus are honorable and they're not going to take it any further. What's to stop somebody else from taking the Keeton decision and as I suggested, getting into a lawsuit to prevent somebody else from doing a compatible language or an API or something else. My best reading of the state of the law and the state of the business climate is: There's no way to prevent people from trying to do that—they will try it and in fact, they will succeed, and we will wind up with a situation which nobody likes.

Ingari: That's not necessarily true that we will come out with a result that we don't like. That could be true for anyone on any side of this discussion. The courts will decide. The decision to pursue the suit—remember what we talked about—first of all the first suit had to do with copying the entire structure of the program. Now you can see that from the external or the internal. In my view the external is more important. Frankly, the user doesn't give a damn how you code it internally; the UI is probably more important. That case has been decided.

The second step is to say, "Here is a vendor who is incorporating your product by subset." That's the statement, whether or not it's proven true in the courts. There is a Lotus menu, a Lotus way to interact with that product. The courts will decide whether that particular kind of inclusion is right or wrong. That process is not a bad process. That process of case law and decisions being made over many years has been one that has served us better than any institution I can think of around here—it's even older than capitalism. And case law works pretty damn well. And it may turn out that Lotus loses this suit, but you know what will happen? That'll be okay because we will all have learned something about how the law wants to come down. And it isn't necessarily going to be an answer that's wrong for the society.

Kapor: The reason that I disagree with the position in this instance is not that I don't believe in the usefulness of extension of case law—it works fine when you're dealing within a conventional, existing, and robust paradigm when you're making incremental extensions to it. But when you have fundamentally new technology whose behavior is so discontinuous from the artifacts of the existing print media that it just doesn't work to try to incrementally extend the existing law, you create enormous distortions and injustices. So my reading of the situation as to why letting things extend through case law will probably result in disastrous consequences is in fact premised on the notion that software is sufficiently different as digital media from other types of media as to create an unacceptably high risk of consequences that absolutely nobody is happy with.

Ingari: You can say the same things about movies and music. Movies and music are not printed media.

Kapor: And in fact the same thing about broadcasting and nobody is particularly (that I know) happy on First Amendment issues. Let me point out that sometimes it works out well, sometimes it doesn't work out well, and we're taking a big risk here, inaction on the part of people who have a stake in this, is very risky.

John McCarthy: I'm John McCarthy; I'm not from any company. I've got about four questions, maybe they've got short answers. The first one has to do with the QWERTY keyboard. Again, because I didn't get a clear answer from the Lotus people, namely would that layout of keys be copyrightable, in your view, today?

Lemberg: Probably not. But one of the problems of the law is, it's true, there are lines, and you can't draw bright lines and it's fuzzy, but the arrangement of 26 letters and, you know, another 10 or 15 symbols, all of which are just out there, that arrangement probably isn't copyrightable. It's a lot different, I think, than the command set of 200-plus terms. But nobody knows because the QWERTY keyboard has been out there in the public domain for so long.

McCarthy: No, I'm sure that nobody could copyright that one today because it's not original. But according to the current law, could Sholes have copyrighted it? And I take it you answered that you believed he couldn't.

Second question: How much would someone have to change Lotus 1-2-3 before you wouldn't sue 'em?

Lemberg: Obviously we cannot answer that but I will say that we've only sued people who copied the whole thing, who copied the whole menus. Borland copied the whole menu. They have one or two, depending on which version, alternate menus, but when you hit "Q123" you get the Lotus menus, the whole things.

McCarthy: Let me pursue that question about "obviously you can't answer that question." If I understand you, and perhaps, I'm inclined to put an unfavorable interpretation of your words, your attitude is "just try it and see; we've got more money than you do."

Lemberg: Until you see something in the flesh you can't tell what it's like. There's a certain aspect of that here—Judge Keeton wrote a decision that I think under the traditional copyright principles is easy: They copied the whole thing, and it was an important part of the product. So you can't look and say, in the abstract, "Copying 42 keys is okay; 76 isn't," or whatever the number is. As I say, we've only sued people who have copied the whole thing. And by the way, certainly we have some money. Adam Osborne was quoted at the beginning of the Paperback case as saying he would bleed us because he had insurance. And actually one reason the case was so expensive to try was some of the defense tactics was just to delay us.

McCarthy: Yeah, I got a couple more. Now I have these retroactive dreams of riches. I haven't really gotten around to dreaming fully yet, but with regard to this copyright of look and feel, one of the feelings of Lisp is to be able to write `(if p a b)` which is only a part of something but it is a sort of look and feel. Ought to I have been spent less time on operation—working on artificial intelligence and to have spent a little more time trying to pick out something like that and copyright it, so that anybody who included **`if p then a else b`** (I could have copyrighted the variants because I knew them all in advance). Should I have pursued that dream of riches?

Kapor: In today's climate, if you were to invent the next Lisp, some new language of today, if you were to go to the office of the outside council of Lotus, if they take you as a client, or Larry Sunseni out in Silicon Valley, or one of the sharks and say, "Hey, I've got this hot language here. This is going to be a big deal! Do we have a case here?", I guarantee you he'd say yes. You might even find an attorney who'll take it on contingency, to represent you, because it's a big thing! Nobody knows because it hasn't been litigated. I'm not a lawyer but I have read Judge Keeton's decision a couple of times pretty carefully and I've thought about exactly that issue. And my belief in the current climate of the law is that if you have an original computer language, you could go for it, based on what Keeton says, with a few extensions, and you'd have a reasonable shot at it. I don't think that that's the kind of climate that we want to have. And if that's the case, the good intentions or the fact that Lotus has only sued companies who copied the whole thing isn't relevant because we're not just talking about Lotus in this case. And that is precisely why I'm so concerned. Because this extension of protection to languages, parts of languages, any imaginable distinctive nonliteral element, the doors have now been thrown open to it. And if we don't do something about it, we're only going to regret it later. If you've got a good language, I've got the names of good lawyers, but I'd rather see that we work together to try to create a situation of maximum group advantage, not individual advantage.

Davis: Okay. Let's move on so we can get through the questions.

AM6: Well, first of all, let me compliment Lotus on surviving something that their employees here just declared impossible, which was that when they started their company, they were competing against a larger company on a very similar product, and they just said that without the protection that you didn't have at that time, you wouldn't have survived. I think you did a really good job because you said that you wouldn't have been able to.

My question is that, when I hear people discussing, "Should this aspect be copyrightable?" or for example, "Should international software be copyrightable across boundary lines?" It seems to be a question of, "Should we take the existing copyright laws and dump them onto this, or should we not?" And that seems to treat the current copyright law as the only thing that you could have done. And that seems a little odd. My question is that wouldn't it be possible that what you need is a different sort of copyright? I mean, maybe it's got the same name, but it's a four-year protection, or it's only double damages or something like that—a novel is still pretty much the same novel after 75 years, so it seems pretty reasonable. But a piece of software that was written five years ago (given the kind of market we've got) doesn't mean a whole lot. So the question is, maybe—do you think that—maybe that we don't need to decide whether copyright applies or not, we just need to design one that fits, what software is, as we've been discussing?

Lemberg: Well, the copyright law has shown amazing flexibility in its well over 200 year history to adapt to many media of expression, ranging from books to films to music and all sorts of things. And it's because, I think, the line between idea and expression is an immensely rich line—you ought not to have protection for ideas, but you do for expressions which are your unique way of doing something when there are lots of different ways of doing something. It's not called expression (and Keeton said this clearly) unless there are lots of ways of doing it. That law has shown itself very flexible through case law. And what we're talking about—a lot of the frustration—is about the common law/case law system that says, instead of trying to have somebody write detailed regulations in advance, which is especially hard in a fast-moving technology, we're going to let these things be decided on a case-by-case basis under very specific facts. A set of facts comes up and then people can read what the judges say about that set of facts, and it guides them. And in fact, in the music industry a long time ago, people didn't know how close you could copy melodies. And now, although I don't know this because I don't do that work, I know that people have a really refined sense about how closely you can copy someone else's melody or rhythm. That's because the case system has worked. I don't think that there's anything about software (and I differ with Mitch on this) that says it can't work here. I think it's starting to happen. We've had lots of people using PC software, for example, where there are more interface issues than anyplace else. Only for a relatively small number of years, and since litigation takes time, we have had very few cases. So I don't think we need a new law; the copyright laws are great. And it's like Frank says; whether or not we win a case, I think it's the right legal method to go.

Davis: Okay. Let's press on, please. I'd like to get through all the questions, if we can.

Ed Bristol: Ed Bristol of Foxboro Company. I deal in software, but software of quite a different application than most people here would be aware of. My company is an old company, it's been involved with patents for many years, so I'm quite comfortable with them and am quite bemused by the difficulties that computer people had with them. I've had a number of patents that I would call software patents, one of which I'll mention briefly later, issued in '72.

Davis: Could later get to be now? We need...

Bristol: I want to make a specific point.

In the '60's, I remember attending an IEEE lecture by a patent lawyer who was describing the difficulties some of his clients who happened to be pioneers in the hi-fi industry were having because the courts weren't supporting their patents. And because of that kind of thing, we lost some of the consumer electronics. Now, I don't believe that the innovation, out-innovate the competition, has really worked. I mean, the semi-conductor industry is having difficulty (that's been their premise) and I think the software industry is going to catch up with them, too. The particular patent I mentioned, I was struggling for about 20 years to get this thing commercialized. We came out with it about '84. That product netted us three industry awards plus three personal awards: an IEEE award, an Instrument Society Award, and an AIChE (Chemical Engineers) Award at about $1000 a crack. A year afterward, Japanese competitors came out with competing products; we shut them down because of the patent.

Carl Hewitt: I'm Carl Hewitt of MIT. It seemed to me a funny thing happened to copyright expression on the way to the computer industry; namely, for the first time, the public started to have an investment in these so-called expressions. Basically, they started to have an operational investment in terms of how they operated. You don't get that out of a song; you don't get that out of a movie; you don't get that out of text. And so, what some companies want to do now in terms of copyright is to protect that, is to appropriate that operational investment. The public has, to their company through copyright. And for example you can consider things like the IBM AT bus which does the equivalent of electronic keystrokes back and forth, and you have to do them in a certain time, and a certain order, in order to make a product that fits on that bus. And if we were to put a tree structure of that command set of the electronic peeks and pokes up on that screen, you'd find that Compaq has duplicated the IBM command set. And so now we have the same thing for Lotus Development Corporation. The public now has an investment in this expression; they've got these little keystrokes, operational keystrokes, built in. And now the question is, can one company appropriate that operational investment of the public to their company in a proprietary way?

AM–anon1: No!

AM–anon2: Good answer.

Lemberg: Well, you know, except there's an investment other people have but the creation of that was not by them, but it was by Lotus.

Hewitt: Same thing for IBM and the AT bus. And it wasn't the appearance on the screen that bothered Lotus so much, but the menu at the bottom.

Kapor: Could I ask Tom a question to follow up on that?

Undoubtedly, the invention was by Lotus, but what is the basis, going back to the Constitutional basis, by which that particular act of creation should get protection? Carl's advancing an argument as to a social good, as to why it should not be protected. We know that Lotus created it; what is the social good in giving that outweighs that factor, is the way that I think I'd phrase his question, and I'd like to know if there's an answer to that.

Lemberg: The answer is that protection of innovation from being ripped off by others, and that's the principal advantage, under the Constitution. Also, the United States is cosignatory to the Berne convention where the moral rights of ownership exist as well. So it comes both ways.

Ingari: There's a fundamental problem in this, though. Are you positing that operational investment is significantly different than spiritual investment? Is it somehow—you can't protect an operational investment, but I presume you wouldn't argue that Mozart shouldn't be compensated for his symphonies, because after all it's just a movement of digits on 50 hands.

Hewitt: What is the public investment, operationally, in Mozart's symphony?

Ingari: There is one, there is one...

Hewitt: ...interacting with that in a certain way that they've built up a training investment for? Did they go to expensive classes so that they could come and listen to Mozart...

Ingari: Yes, they do. You got any kids in the New England Conservatory?

I'm sorry, but I think that that's a very arrogant and elitist distinction to say that the—hey, I'm sorry—I was a musician, okay. I tried to make a living as a songwriter. So don't tell me that songwriters don't deserve protection but operational investors do. I don't buy it.

Hewitt: What is the input from the public to a Mozart performance? What investment do they have in it?

Ingari: You choose to make an investment in learning a product or playing a symphony. No one has forced anyone to buy 1-2-3. I don't buy that argument. Who forced corporate America to buy 1-2-3?

Kapor: It's a little more complicated than that, because while no one forces anyone to buy it, people are forced to use it, because of the widespread standardization at the corporate level, which has been very good economically for Lotus. You've got corporations; they standardize on it, and users are in fact forced to use it, because it shows up on their desk along with WordPerfect, or whatever the standard software is there. And there is a social investment when something becomes a standard at that level. That's just a fact; I'm not sure exactly how to grapple with it. But I believe the point that when there is a public standard that's transcended the economic interest of the individual creator, simply saying that creator's rights should be the sole determinant of whether there should be some protection or not is not a good strong argument.

Lemberg: Well, it's not the sole protection, but it's hard not to throw the baby out with the bath water. If you protect innovation except where it's successful enough that people want to copy it then you haven't done something terribly meaningful.

Ingari: I find that bizarre. This is an argument for all technology driven industries to remain small. This would dismantle every great industry that's grown out of this institution. This would say the pharmaceutical industries would all top out at $100 million; the chemical and process industries...When you achieve standardization, therefore everything opens up? I don't get it.

Davis: Hang on one second. I'm going to invoke my privileges to continue us. Please.

Kent Pitman: Kent Pitman, Symbolics. Like everybody else here, I don't speak for my company, either. I was caused to wonder as I was standing here in line, if maybe we couldn't turn the lawyers on themselves and have them do look and feel suits about that particular suit so that maybe they would bog down their own profession.

Davis: Lawyer jokes test the boundaries of civility. Please...

Pitman: Sorry—I didn't mean that as a personal attack...

Davis: This gentleman came here this evening out of his own free will. Please respect that.

Pitman: But it does actually relate to the point that I wanted to make, which is that drawing a distinction between an engineering and a creative aspect of computer science, and it's not one that we really touched on a lot tonight. The way that computer science is traditionally taught, you go into a room and someone tells you all about the constructs and how they are supposed to be used. And then you're told to go out and write a program. And you're going to get graded, typically, on how closely your program matches the program that the person who assigned you the project expected it to. He's not likely to come back and say, "No, that looked just exactly like what I wanted you to do. Why don't you go back and do it again?" And in fact, within my company and only on a very informal basis so far (although it makes me very concerned), there have been times when I've come up with ideas relatively independently (obviously I'm familiar with what's going on in the industry) where I'll say, "Wouldn't it be nice to have a menu item that does such-and-so?" and someone will say, "Well, such-and-so other company does that. Maybe we should rename it or make it different." And I think that there are a lot of dollars that are being invested right now in coming up with things that are basically just gratuitously different. I think it's in a way fortunate that the precedent that we have for Lotus 1-2-3 is that we have a menu item which, I bet if we took it out to the real industry and we said, "What kinds of commands would people want? And how would they want them organized?" Statistical data would not say to lay them out as 1-2-3 did. But I think that it should be appropriate for two independent organizations to go out and do field data and it would not be surprising if in fact they had a very complex graph that came back the same. The odds would probably not be 100 percent, but they would be within the realm that we wouldn't find it unusual. And I don't think that we should have laws that work against that. Somehow we should take that need for—I think that's what Carl was getting at when he said public investment—is that need for standardization. There are only so many ways you can draw a rectangle and I don't want to see interfaces skewing to the side a little bit or monitors putting the pixels offset. And we have to develop laws, in this particular case that are going to work towards having interfaces where we can get that community again.

Davis: Any comment on that from the panel, or do you just want to press on? Okay, then why don't we just press on.

AM10: I just wanted to raise a couple of questions about how you develop a—I guess what people are trying to say—how you develop some sort of general policy or doctrine, rather than just case by case randomly stumbling through this. I happened to hear a lecture earlier this week by a professor at the Harvard Law School addressing a bunch of students on the question of copyrights in music, which I guess you referred to, Mr. Ingari. And I was discouraged listening to this because what he basically said, if I understood him correctly, was there were two major cases that are always cited as precedents, precedents for copyrights in music. He thought both of them were logically invalid, but he was telling the students you have to use these, nevertheless, and here are some clues about how you structure your case, in a given case, so that you can use these to appease the judge, because they'll always refer to these mentally; you have to refer to them somehow, but use them so that you can use the useful parts of them, and ignore the fact that they are not logical. Then he also made the point that, he sort of said, as far as I can tell there is no doctrine. It seems to be decided on a very fact-specific basis for every case of copyright in music, and generally the better known performer wins. Mick Jagger was either sued by or suing somebody—I can't remember what—and he cited similar cases and basically, the well-known guy would spend three hours playing his music in court, everybody would love it, the jury would decide in his favor. That was what he concluded. I don't know if he knows what he was talking about; this was a professor, Harvard Law School.

Davis: Bounds of civility exceeded! Ten yard penalty.

AM10: That was the specific point. The more general question, that is, what degree of protection for intellectual property is required: I think that is the general question. What degree. And I would say, if you take the American Constitution and not the Rights of Man definition, and worry about promoting science and useful arts and all of that, the barrier to entry is the financial risk of bringing something to market. For a long time, people did not bring into useful medical therapy things that came out of NIH sponsored research because that was legally forbidden; and that changed sometime in the late '70s or '80s. And then all the universities started having offices for technology licensing.

Davis: You need to get to the question here.

AM10: So the question is, could you develop a standard or degree that would be tied to the financial risk involved in bringing something to market? In pharmaceuticals it's very high; you need patents. In software, if it's low, maybe you can argue that you don't and have it a spectrum in between.

Lemberg: Can I comment on the comment about the law professor? I don't know. You know, there are a couple of things to say. One is that all human endeavors are imperfect. I don't know who the professor was; I don't know the cases; they may be without logical basis. But it's often very easy to look back on something and take the cynical view that this is all a crock of stuff. Now this may be and it may not be. So there is a view of all human endeavors that's very cynical. This person seems to have that. That doesn't mean that a system of case law doesn't provide some guides and that guides don't have some fundamental principle. You can usually attack legal opinion (they're often long) and find some hole in it, even if it's a very good one. That's true for works that people in this room have done, of course, even if they are great works.

The other thing is that we live in a system of law, like it or not, and the notion of if we ask Congress to rewrite this statute (the copyright laws are very general), we ask Congress to write a detailed law regulating the intellectual property protection for software, if you think that process is going to be more intellectually rigorous and make everybody feel happy about it in the end, I suggest that you go to Congress some day and watch it function.

Kapor: Yeah, I would say if you just turned it over to the Congress critters, you kind of get what you pay for. But about ten years ago, actually maybe a bit more, Congress convened something called CONTU, Commission on New Technological Uses, which attempted to grapple with the whole set of issues involving copyright in new media. I'd suggest that a time is rapidly approaching, sometime in this decade, for the reconvening of CONTU, or a new CONTU, because what's really needed is to get out, get the debate out of the hands of this very adversarial litigation because that does not tend to produce any new insights. We have a situation where there is a sufficient novelty in the technology and its uses that I think a Congressionally sponsored commission to try to begin to sort out what's to be done in the long term would be the proper follow-on to a short-term holding action of pulling back to where we were a few years ago.

I want to second the notion that was raised here by a commenter a couple of minutes ago. There is a real cost to proceeding in the fashion that we are proceeding with case law, and that is the chilling effect of the current environment, already, on the software developers, all of whom, I know, are nervous about contemplating the inclusion of a feature or set of features in a new product which resembles the features of an existing product. Even though no judge has said, "You can't do that!" it becomes more and more plausible to believe either that some judge will decide that you can't do that, or more likely, some company will come along with an agenda to persuade some judge of that, as Apple is absolutely famous for. And you have people in the software industry spending more time worrying about how not to get sued, than they have in investing and making their product. That is a terrible side effect, and the longer we go in the condition of legal uncertainty in which we're in, as to, "What's going to happen next; which domino is going to fall over?" it's a terrible thing.

Davis: Since we're running fairly beyond our scheduled time, let me suggest we take one more question from each side [of the room] to finish up, and then we will keep around the panelists as long as they're willing to stay and run this informally. My apologies to those of you that were further back in the line.

Nate Osgood: My question concerns the fact that throughout this session there has been a lot of theorizing (it seems most of it is very hypothetical), about what would be the effects of changes in copyright laws; or for forms of copyright laws, some people say that they would stifle innovation, others that other forms of the copyright laws might lead to reduction in the industry size, permanently limiting it to very small companies. My question is this. We've also seen a large amount of discussion about international issues, about different countries that have taken different approaches to copyright laws. I was wondering: It seems to me we have the potential for a wide variety of experimentation with different copyright laws, and the ability to see the results of those. Is there any one of you who could comment on the effects in those particular countries where copyright laws are different, what the effects on the industry have been, or is there not enough information to generalize at this point?

Lemberg: It's a little hard to generalize. One example is Brazil, which finally, although they're fighting against this notion in the international tariff negotiations, adopted a copyright law for software about two years ago. One reason that they did is that their fledgling software industry demanded it. They felt they couldn't compete if they didn't have basic protection.

Davis: Next.

Mitchell Charity: Along what timelines do you see this dispute evolving?

Landry: Dispute evolving?

Charity: Some issues like the condensing of the European market or the deed stabilization of the software industry based on suits can be—you may have some handle on what timeframe we can expect to see such effects.

Landry: The whole time.

Kapor: I would say that it looks like a contentious decade because we haven't even begun to see the patent wars yet, and it's difficult to see more than ten years out. So as far as we can see, I would say it's going to be a difficult issue.

Landry: Just one comment on the patent wars, since the patents were brought up and one gentleman here talked about how he got his patent and how he prevented the Japanese. I got a document today which was the top 200 companies for patents issued in 1989. This is not software exclusive, but there are a number of software patents being granted to these companies. Number one is Hitachi, Limited; #2 is Toshiba (these are US patents); #3 is Canon; #4 is Fuji Photo Film; #5 is General Electric; and #6 is Mitsubishi. So five out of six patents being granted in the top six companies are to Japanese companies, and that's what has a tendency to worry me about the patent laws and software right now.

Davis: We're going to do one more quick question. Yes, sir, will you please identify yourself.

Richard Bezjian: I'm the President of Mosaic Software.

Davis: Let me point out that this was not a set up, he just wandered up here at the end.

Bezjian: And by the way, we are still in the lawsuit, and indeed the lawsuit does bring crippling effects to the company's business and its development. (If anybody has any idea of any kind of help, we would appreciate it.)

I do agree with Mitch that software technology is, perhaps, significantly different than most of the things we have seen in the past, and therefore, there is an inherent problem of how to twist the copyright laws that exist today to fit this new technology. And in that regard I just want to mention to you that we at Mosaic software did in fact go to great lengths to develop the source codes for our products (Twin, Integrate 7) in a significantly different language (C and 1 and 2 PS), as far away from 1-2-3 source code, and so on and so forth—we went to great lengths for that. That in regard to the user interface. And I must tell you that the development of the Software program, maintaining it, keeping it updated, keeping it documented, porting it to other computers, evolving it, enhancing it, supporting it, answering service calls, and so on and so forth, this is the bigger picture, and it is really simplistic to say that if one can actually just take the looks of a program, they will be able to run with it and in fact surpass whoever originated the first program. The software technology in itself has inherent differences from that aspect.

My other question is that, to all four of you, these kinds of rules that are coming up now have all of a sudden made a lot of people—a lot of very honest, a lot of very straightforward developers, software engineers, businessmen, who in the years of '75 to 1985 did a lot of development, did a lot of programming, and brought together what they thought was a very good program—these new rules are all of a sudden declaring all of these people (and there are a lot of those people) in fact have done a very illegal thing. What is your response to that?

Lemberg: I think under traditional copyright principles, if you looked at them in 1983 or '82, or picked some other date, the kind of copying, the total copying that we sued on was pretty clearly a violation. The notion that somehow the only thing that was protected was the code and not the stuff that the user sees is an idea that sort of crept into people's minds, but it didn't creep in there from the law, the history of the law and other media, or or anything about the cases that were decided early on about the software.

Landry: I don't know why you continue to trivialize the work that goes on. Frank mentioned that he said that really the whole thing is in the user interface and that there's nothing that goes on in the code. . .

Ingari: I never said that...

Landry: All right, all right, I'm extending your point...

Ingari: That's a fairly sleazy attack, and I would appreciate it if you would be a little more accurate...

Landry: Well, what did you say? Maybe you can clarify your statement for me, but, in this particular case...

Ingari: I said that the process was integral, and that work on the internals is connected to what is made visible to the user; and that the division is getting fuzzier. I never said that it was all in the UI, thank you very much.

Landry: You're welcome very much.

Ingari: It's very convenient to have Lotus down the street as "the big company." It's an important thing for people, especially—there are a lot of you out there I think are in the student domain, though not that many—just to do a little bit of projecting: This law and the application of the law is important no matter who is in the roles. What if Lotus was the small company in this situation. You can laugh at it—Tom Lemberg and I will pass. Lotus will pass. These issues are larger issues. It's very convenient and easy to have a big target to poke at, but try to think about the underlying issues and not to make the people and their motivations—maybe we are arrogant, maybe we make too much money—the fact of the matter is that the issues are real. And try to take the personalities out of it as you think about it, and not succumb to the easy applause for simplistic attacks.

Landry: Well, I wouldn't agree with that at all. The point is that—I worked at McCormick and Dodge, which was 1600 people, and at Cullinet, that was 2700 people. So I don't think I really come from a small company background exclusively. The point that I'm making is that my belief, as I've said before, is that if Lotus wants to compete in the marketplace, there are better places to do it than in the courtroom. And the fact of the matter is that there is a significant amount of work and investment that went in by Mosaic and Paperback to build a system, only to find out that they were going to be sued—and sued, in fact, successfully—on something that is still very (obviously, by this very discussion) very debatable as to its accuracy. Lotus could have, in fact, innovated a better product, and in fact made that issue somewhat nil.

Lemberg: There's no separation between innovation which is absolutely anybody's duty if he's going to continue and to succeed, and no copyright, set of copyright victories, can insulate any of us from failure if we don't do that. There's no reason why we can't innovate and sue at the same time. Whether or not...

The amount of money you spend on a lawsuit is a lot less than we spend on innovation and development and if we hadn't we'd be in a lot of trouble. If the market says that we haven't innovated enough that's a problem we have or any other company would have. But it's another thing to say you should stand by and let someone rip you off.

Kapor: In the interest of concluding on the note of civility which permeated at least 98 percent of the evening: I would encourage people also to keep a clear separation in their minds between the personality and power dynamics of the situation (it's a free country; everyone's entitled to their opinions) and the underlying issues. They come together in very interesting ways, one of which, I've suggested, is that people tend to take up points of you that further their self-interests and wrap around them arguments as to the principles. I think that that unwrapping is something that is just as necessary to do in the case of your evaluating what I have to say as to what all other parties say; but I suggest that it's really important to get to the underlying issues, and I am not persuaded, as carefully as I have listened to the case, that maximalist protection furthers innovation, that those arguments hold water. We're going to continue debating it, but to the extent that you come to a similar conclusion I'm going to suggest to you that a time will come when it will be necessary to move from contemplation to action because we are going to have to go to the Congress about this if we want to get it changed.

Davis: Thank you. And with that let me call a halt to these proceedings with just a few last comments. I can't resist picking up on Mitch's comment earlier on that he felt a bit like Frankenstein. Remember, the book read carefully demonstrates that though the monster is large and extraordinarily powerful, he is fundamentally benevolent, and it is in fact Dr. Frankenstein who comes up short on the moral dimension.

AM–anon3: Civility rule!

Davis: So choose your metaphors carefully.

Anyway, it would be extraordinarily hard to try to summarize what happened tonight so I won't even try, except to say that we've got the issues of narrow vs. broad protection; and I think an issue emerged here tonight that I really hadn't anticipated, which is the question of what even is the process to be? We had arguments that the case law process is going to work its way out and that's the right way to let it happen, and we had arguments that this stuff is fundamentally different and that process isn't going to work; you have to go about it differently. So I think added to the agenda of things to think about is not only what is the answer, but how are we going to even get there? What's the process by which we're going to get to something that we're all going to want to live with, that's going to produce the kind of world we're going to live in?

Let me finish up by thanking all of our four speakers, particularly the folks from Lotus who knew that they were wandering into the lion's den tonight. I think it's been an intriguing and fascinating evening. Thank you all very much.

This transcript can be obtained by anonymous ftp, and a videotape of the panel discussion is also available as AIV-7. Contact the MIT Artificial Intelligence Laboratory's Publications Office for details.

Publications Office
MIT Artificial Intelligence Laboratory
545 Technology Square
Cambridge, MA 02139
(617) 253-6773
email: publications@ai.mit.edu
fax: (617) 253-5060