A.2.5. Legal Issues Related to Software

A well-defined policy and guidelines regarding the acquisition, development, distribution, and support of computer software must be conceived concerning technological, legal, social, and marketing considerations.

Many problems exist, with potentially significant and costly legal implications, regarding the acquisition, distribution, and utilization of software products. Among the most common issues, mention is made of: the use of illegal copies of commercial packages; disregard of copyright issues and ownership rights in contracting external software development; inappropriate agreement provisions with contracted external systems developers; lack of guidelines regarding the selection, testing, and evaluation of software products, and the lack of guidelines regarding the transference of software products.

A.2.5.1. Some Relevant Definitions

- "Source code" means the set of instructions as written by the programmer in one or more computer languages.

- "Object code" means any instruction or set of instructions in machine-executable form, also referred to as "compiled code". Object code is generated by processing the source code with specialized programs called compilers.

- "Technical documentation" means any printed, magnetic, or optical media material containing a detailed description of the internal organization, procedures, and sequencing of machine operations related to one specific computer software product and should include a copy of the source code.

- "User documentation" means any standard manuals or other related materials used for user instruction or reference in use of the licensed program.

- "Licensed program" means the object code version of the program and related program user documentation.

- "Use" means copying of any portion of the licensed program from a storage unit or media into the designated equipment and execution of the licensed program on that equipment.

- "Copy-protected software" does not, as it may appear, relate to copyright regulations, but to the existence of software or hardware solutions that make it difficult or impossible for users to make copies of the object code or prevent installation of the programs except for a pre-determined number of times. There are many copy protection schemes as well as many ways to by-pass such limitations. Much used in the past, presently very few commercial programs

still use such resources.

A.2.5.2. Trends in Software Design and Developer's Rights

In the first-generation electric tabulators and electronic computers, the instructions ("software" or "programs") were hard-wired by the physical connection of rows of contacts. Because of the desire to be able to run multiple unrelated applications without the bother of rewiring the contacts every time, removable panels with wiring were later introduced. This approach was later improved by the introduction of the "loaded-program concept", using removable boards with appropriately connected cables and plugs similar to the ones utilized by early hand-operated telephone exchanges. Subsequently, decks of punched cards, initially only used for data entry, storage, and loading, were also employed for loading the instructions to be used by the electronic circuitry to execute a defined set of sequential operations.

Eventually, magnetic media of varied formats (disks and tapes) and optical media (laser disks) became the support elected for general use in the distribution of software and data storage. Up to about a decade ago, commercial programmers relied on secrecy and on the strict enforcement of copyright laws to ensure competitiveness of their programs. The arcane command structure of early programming languages also provided a significant barrier for outsiders to copy or emulate such early products.

Based on the origins of programming, patent lawyers argued that the software was just another way of wiring up a machine. This concept deemed the duality computer-software not much different from any other piece of equipment and thus free to be patented. By 1995 some 12,000 software patents were been issued, and about 3,000 more awaited review. Software developers found trespassing on other firms' intellectual property have been sued and large sums have been paid as settlement for such litigation cases.

More recently, the trend has been to get away from patenting and move towards the implementation of strong copyright protection and its enforcement. The concept, however, is fully valid only for what is usually labeled as "generic" or "development" software. Such products, usually programming languages, are employed in the development of "applications", i.e. software written in one or more of the generic products with the object of performing a predetermined sequence of data manipulations with a well-defined purpose. The burden of proving the occurrence of copyright infringement is much more complex in the case of an application software product.

From the point of view of patent rights, in contrast with the generic products or development programming languages, an application software is built out of thousands of lines of source code that rely on standard expressions that are part and characteristic of the programming language syntax, rules, and conventions (e.g., sort values, select the larger of two values, compare a set of variables, write to the video monitor, or transfer to a printing device, etc.). Similarly to what happens in literature, it is the way known ideas are expressed — not the ideas themselves — that makes a computer application program useful. Here, the issue becomes the protection of the intellectual production of software developers, in a manner not dissimilar to the one afforded to writers, musicians, and movie or video directors. Although the issue of software property rights is still very cloudy, the prevailing

view is that such a computer program is entitled to copyright "as a literary work".

Radical innovations in the way applications were developed occurred in the 80's. That decade witnessed a confrontation between the traditional role of the Data Processing Department, as the omnipotent master of information technology, and end-users who could, for the first time, afford to buy their own systems, including development software, and design and write their own applications. The growing power of desktop computers and user-oriented languages empowered the users to turn data into information, information into decisions, and, through communication, decisions into knowledge. However, the tools of the 80's were crude and cumbersome to use, and the deployment of systems was limited by cost considerations, the slow learning curve, and the quality of the applications. More recently, with the introduction of a large variety of productivity tools designed to work in conjunction with many of the available programming languages, the property rights situation of applications software became a great deal more complicated. Those highly efficient tools assist programmers in the technical definition and documentation of systems and in the generation of programs ("application generators"). They can produce code that will control everything in a program, from the code used to drive the data-related procedures and reporting to the construction of the user interface with all the “bells and whistles” that may be desired.

In the development of applications, the "programming without programmers" became a reality. Over the next decade it is expected that open standards will become generalized and user interface and programming tools will evolve to become more and more user-oriented. They will incorporate cognitive elements such as object-oriented programming, authoring tools, extensive use of neural networks and knowledge navigators, integration of multimedia and systems built on small, portable code, with many layers of high-level interfaces. Object-oriented technology will allow greater freedom to assemble skeleton routines or components into a solution that could be tailored to very specific implementation environments.

While traditional development focused on the optimization of the application for the most efficient use of the hardware, next-generation development software will continue to emphasize optimization based on users' needs, even when this may require larger programs and more lines of code. The source code of routines of different applications, generated by software development and productivity tools utilized by different developers, will certainly have large portions that may be identical. In these circumstances, the question of the confines of the developer's intellectual property becomes more difficult to ascertain or justify for a specific product.

The appearance of multimedia created new, and broader, legal enforcement problems. The combined use of a scanner, sound board, and a multimedia authoring software allow a user to incorporate a large quantity and variety of data from the surrounding environment. It is rather improbable that someone will end up being prosecuted for digitizing and using a "stolen" segment of music or a scanned picture from, for instance, the National Geographic Magazine in his or her microcomputer production, although the fact remains that the contents of virtually every book or magazine are copyrighted, as are films, TV programs, tapes, CDs, etc.

A.2.5.3. Ownership Rights

Software ownership and attending rights fall into two categories: *proprietary* and *public domain*.

Proprietary Software

"Proprietary software" corresponds to products, usually developed for commercial purposes, where copyright or, more rarely, patent protection applies. Proprietary software can be distributed through a variety of outlets: computer manufacturers, software houses, retail software businesses, and special distribution schemes as is, for instance, the case of "shareware".

Shareware distribution gives users a chance to try software before buying it. If you try a shareware program and continue using it, you are expected to register and pay for it. Individual programs differ on many details — some do not need mandatory registration, while others require it, some specify a maximum trial period. With registration, anything from the simple right to continuing use of the software to an updated program with printed manual could be had. Copyright laws apply to both shareware and commercial software, and the copyright holder retains all rights, with maybe a few specific exceptions. Shareware authors are accomplished programmers, just like commercial authors, and the programs are of comparable quality. The main difference is in the method of distribution. The author specifically grants the right to copy and distribute the software, either to all or to a specific group of users.

Commonly the user of a proprietary system is licensed for one or more installations of each product. A "site license" is a relatively inexpensive way for more than one person to legally use one copy of a program on more than one computer at a time. Site licenses are designed for companies, offices, or workgroups where more than one person in the organization needs to use a product. A site license does not require the acquisition of additional original installation disks.

All programs are distributed as a licensed product for use and very rarely sold to users. A licensee understands and agrees that the source code for the licensed program and all related documentation constitute property and trade secrets of the developer, owner of the copyright to the licensed program, embodying substantial creative efforts which are secret, confidential, and not generally known by the public. A licensee usually is supposed to agree that, during the term of the corresponding license, he will hold the licensed program, including any copies and any documentation related to it, in strict confidence and not permit any person or entity to obtain access to it except as required for the licensee's own internal use. It is required, under the law of most countries, that the licensee shall inform the developer promptly and in writing of any actual or suspected unauthorized use or disclosure of the licensed programs or related documentation.

Public Domain Software

This type of product is distributed free of charge although the developer may charge the cost of the distribution media, printing of documents, and mailing. There are a very large number of products in this category and many specialized catalogs of public domain software are regularly

published. Many international, governmental, and non-governmental organizations have been instrumental in developing and distributing public domain software and the tendency has been to have products developed with public or international resources to be considered in this category.

A.2.5.4. Understanding Software Copyright Protection

The licensee of a copyrighted product is granted a nontransferable, nonexclusive right to use an agreed number of copies of the licensed program. In general, the licenser will deliver one copy of the licensed program to the licensee. The licensee may make additional copies of the licensed program, up to the number of copies licensed, provided that each copy of the program contains the developer's copyright notice and any other proprietary legends deemed appropriate and necessary.

In general, each copy of the licensed program provided under a license may be used on only one computer at any one time. If used on a network system, each terminal user is frequently considered to be using a distinct copy of the licensed program whether or not he is actually using it.

It is understood, under copyright protection agreements, that the licensee shall not use, copy, rent, lease, sell, modify, decompile, disassemble, reverse engineer, or transfer the licensed program, except as provided in the pertinent agreement. Any such unauthorized use may result in immediate and automatic termination of the license and eventual legal prosecution. A license is effective until terminated unilaterally or by failure to abide by the conditions of the license agreement. The licensee may terminate his binding agreement at any time, by destroying the licensed program and all copies of it. In some cases, he or she may be required to notify the developers in writing and, on termination, the licensee is supposed to erase or destroy the magnetic media containing the program and may be bound to return part of or all materials not destroyed to the developers, together with a written declaration that the eventual remaining materials have been indeed destroyed.

In the past, the extent of software copyrights was interpreted as protection to the source code, any translation of the source code into another language as well as the structure, sequence, and organization of the source code, including routines and subroutines and the order in which they are called by the sets of instructions written by the developer. The protection of the structure and organization of a software product is equivalent to the protection against plagiarism awarded to the plot or ideas in a written work, even if the actual text may not be identical.

Cases have been brought to court in the U.S. and the U.K. where these matters were taken well beyond copying the structure, sequence, and organization of the program code itself — and copyright infringements have been found when programs do no more than exhibit close similarities at the user level interface, even when the structure of the underlying code is completely different. The commercial computer software industry has long been advocating that the user interface should be entitled to copyright protection.

With the appearance and widespread utilization of application generators and other software development aids, the question of protection of the structure, internal organization, and user interface of programs becomes, however, very difficult to justify. Much controversy around this matter is expected to occur in the future.

A.2.5.5. Copyright Issues in Contracted Software Development

It has been a general understanding that in the absence of any specific assignment of ownership and copyright to the customer, the product development work done for a customer by someone who is not directly employed by the customer carries limitations on the contractor's ownership rights. In this situation, the ownership and copyright rests with the developer, independent of the fact that the developer may have given the contractor an unrestricted right of use of the developed product. Even if the commissioned work represents just the rewriting of portions of an existing program or making improvements, the contractor may be faced with a property claim by the developer.

Too many instances occur of contracted software development carried out without proper attention to the above principle and organizations have found themselves, with products that they paid for, without technical documentation or access to the source code. The caveat here is that a written agreement, dealing specifically with copyright ownership, must always be secured before development work is initiated.

Similarly to the situation observed with the introduction of photocopiers, which created a generalized and mostly uncontrollable infringement of copyright regarding printed works, the ease of copying magnetic media has facilitated the illegal reproduction of copyrighted software. It has been claimed that in some countries 90% of generic commercial software in use consists of illegal copies. Enforcement is a major problem and there are many who believe that copyright protection of software products should be revised considering the realities brought forth by modern software development tools and the prevailing attitudes and practices.

A major problem in many organizations, and potentially costly from the legal point of view, is the rampant infringement of copyright requirements for proprietary software. This situation emulates the also generalized illegal copying of software in private computers. Users and managers have become too complacent with the practice and this may put their organizations in an exposed situation if the software developers or legal owners decide to prosecute. Many episodes of legal action taken against organizations in the U.S. and other countries should be considered as a serious warning.

A clear definition of the responsibilities of the organization and the introduction of controlling mechanisms to deal with the problem of illegal software utilization must be addressed. It must be clearly established that proprietary software acquired, installed, or distributed by an organization should conform to existing legislation, and efforts should be made to enforce such regulations. The responsibility to establish and enforce such actions should be left to the information systems manager or, when one does not exist, to the chief executive or manager.

As a distributor of proprietary, commercial products or in the case of transference of products developed by other organizations or individuals, the organization assumes the legal liabilities related to improper utilization as well as the responsibilities regarding product registration and dealing directly with the developer or supplier, if eventual problems do occur.

A.2.5.6. Developer's Responsibilities

There are technical and ethical obligations of the developer or legal owner to support the user and provide maintenance of the software for a reasonable period of time. A few pertinent points must be stressed:

- A developer, even in proprietary or commercial products, never warrants that the program is free from coding errors. Usually programs are distributed or licensed "as is". However, program problems reported to the developer and determined to be actual coding errors should be corrected within a reasonable time.

- The limited warranty that usually accompanies a program also does not apply to the extent that any failure of the licensed program to perform as warranted, if the program is not used in accordance with the user documentation or modified by any person other than authorized personnel.

- If existing, the liability of developers or suppliers for any claim or damage arising out of the use of their products is commonly limited to direct damages which do not exceed the license fee(s), which have been paid by the licensee for the specific product that is the subject of such claim or damage.

- It is a standard clause in agreements that no developer is liable for damages, including lost profits, lost savings, or other incidental or consequential damages arising out of the use or inability to use the software product in question.

- In transfer or license agreements, the agreement or any rights or obligations cannot be assigned or otherwise transferred by the licensee to other parties without prior written consent of the developers, providers, or distributors.

Proprietary software developers maintain great secrecy of their trade products. There are, however, many possibilities of obtaining source code from independent developers and other sources. Most developers, even of non-commercial products, usually do not share freely the code of their applications. In general it is not advisable that the source code and complete technical documentation, even in the case of public domain products, should be made indiscriminately available. The responsibilities of the developer regarding software maintenance and upgrading and the interests of maintaining standards of data definition, programming routines, reporting, and quality may be impaired by the proliferation of many "copy-cat" versions of the original product.

In very specific circumstances and after assurances regarding the technical qualification of the recipient and clear definition of intended use, source code and technical documentation may be provided to other parties with the clear understanding that the alterations will be the total responsibility of the new owner. Although credits may be given, the changed products should be clearly differentiated from the original ones to avoid confusion to the end-user.