DRAFT

SOFTWARE POLICIES

FOR

DEVELOPMENT AND EXPORT

Economic Development Division
Industry and Economic Development Branch
Technology and Industry Sector
Department of Communications

October 1984

DRAFT
BROUILLON

SOFTWARE POLICIES
FOR
DEVELOPMENT AND EXPORT

a research report
for
The Department of Regional Industrial Expansion
and
The Department of Communications

by
Russel M. Wills
Peter J. Booth
Steven Globerman

WESCOM COMMUNICATIONS STUDIES
AND RESEARCH LIMITED
105-853 Richards Street
Vancouver, B.C. V6B 4K5
(604) 669-7175

August 1984

DRAFT

BROUILLON

TABLE OF CONTENTS

DRAFT

BROUILLON

TABLE OF CONTENTS
(continued)

DRAFT

BROUILLON

TABLE OF CONTENTS
(continued)

DRAFT

BROUILLON

TABLE OF CONTENTS
(continued)

DRAFT

BROUILLON

TABLE OF CONTENTS
(continued)

DRAFT

BROUILLON

TABLE OF CONTENTS
(continued)

ACKNOWLEDGEMENTS

We wish to thank the firms and governments whose personnel gave generously of their time and information and to thank the Departments of Regional Industrial Expansion and Communications for their roles in coordinating this work. The descriptive sections on educational computing were written by Teresa Plowright. Finally, we wish to thank the International Development Research Centre for the use of material concerning software policies of Brazil and Singapore.

DRAFT

BROUILLON

1.0 INTRODUCTION

This work examines policy instruments for Canadian software development and export in the short, medium and long term perspectives, with emphasis on the short and medium term (two to five years).

Because (1) there is a considerable consolidation among North American software firms producing traditional business software, and (2) the production of this type of software is particularly amenable to being partially automated and produced cheaply offshore, from the immediate perspective there are two imperatives: taxation/fiscal instruments which effectively support the Canadian software industry in its present form, i.e. being mainly involved in the production of traditional business software, and initiatives and instruments which allow firms to make the expansion to new product lines with a minimal amount of fiscal disruption.

With respect to critical economic indicators such as the balance of payments in high technology goods and services or the extent of foreign control in crucial industries such as computers and software, Canada more resembles some of the newly industrialized countries (NICs) such as Brazil, Singapore and South Korea than it does OECD countries such as the US or Japan. And yet our policy scope for both technology strategies and specific fiscal/taxation support measures are usually limited mainly to those of OECD countries.

It is now apparent that much of the production of common business programming, comprising more than 80% of all world programming -- will soon be packaged and semi-automated and that its production requires comparatively low-level skills. Some countries have correspondingly instigated fiscal, export and tax incentives to technologically upgrade their national software production away from these traditional business applications into areas such as expert systems and non-procedural query languages, applications generators, process control (CAD/CAM) software for industry, natural language processing, etc. Several elements of such strategies and fiscal, tax and export incentives are of applicability to the Canadian software industry, which is comprised mainly of firms producing simple business applications or marketing US software packages.

It is also apparent for a number of reasons, such as the trend toward software embedded into computers, that software policies are intricately tied to those involving hardware.

1.1 Software Industrial and Business Trends

On the basis of data from Phase I of the Department of Communication's 1983 industry study,[1] other recent software studies such as the multi-volume Cognos study,[2] Wescom research and firm-level data collected by the authors, this work first delineates current and future software and hardware products and industrial/marketing trends of the 1980's. A knowledge of such trends are prerequisite for meaningful policy/strategic consideration.

1.2 Industrial and Policy Comparison

As further preliminary analysis, this work also compares the Canadian software industry with those of the US, Japan and (most relevantly) selected NICs previously mentioned, Singapore and Brazil. This comparison is made with respect to:

- comparative constraints (i.e. nature of the software workforce, training necessities, comparative rates of industrial automation, etc.)

- trends in the world software industry (such as future offshore production of business applications by the developed countries)

- export, fiscal, tax, R&D, manufacturing and marketing support given by national governments to their respective software industries, etc.

1.3 Policy Instruments

Lastly, this work examines and evaluates specific policy instruments for Canada.

In Canada numerous firms have emerged in the last three years focusing their efforts on the development of computer software. These companies are essentially of two types -- those developing mainly custom but also packaged software and those marketing applications packages produced offshore, mainly in the US. In spite of the considerable time normally given over to software maintenance compared to production and development, Wescom statistics indicate that until recently the marketing and maintenance function has dominated the Canadian software industry with a decreased emphasis on the development of new applications. Many firms, which tend to be of a limited size, are serving as marketing agents for US firms, which are rapidly dominating the low end of the business applications market and are providing relatively trivial packages for accounting, word processing and numeric analysis. Examples include the Visi Corporation of the US, which produces the highly successful Visi series of applications packages. Others include Peachtree, which is developing software for the IBM PC, and companies such as Microsoft and JD Edwards, specializing in accounting and financial packages. Many Canadian firms provide the initial sale of such offshore-produced software packages to business and derive further revenues from the support of these packages.

Developing an increased Canadian presence in a worldwide software industry requires that efforts be made to encourage Canadian firms to direct more investment to higher level applications in addition to support and maintenance.

The production of higher level software involves products in communications, CAD/CAM, network control, navigation and mapping, computer learning, resources exploration and processing, applications generators, and virtually all other AI-based applications such as expert systems, industrial robotics, natural language query of data bases and vision and tactile modeling. Our preliminary studies show that some of these areas, properly developed and supported, could provide a base for larger Canadian exports.

But given the substantial support lent to software industries by some of Canada's competitors such as Japan and the NICs, it is clear that some recently proposed industrial support measures for software are simply not sufficient. (These latter include such measures as expanding the definition of R&D to include software design and writing and then expanding R&D initiatives to include marketing and pre-production costs of software -- also that sales tax exemptions be granted to software and that favorable taxation measures -- such as prorating equipment used in software writing, manufacturing investment tax credits and accelerated depreciation allowances -- be applied to the software industry, etc.)

Given the support lent to software industries by other nations, these measures may not be good enough. Fiscal/tax, export and other support measures examined involve:

- accelerated software training programs in the NICs (i.e. the Japan-Singapore Institute of Software Technology and Singapore's programs for rapid software education)

- software export support and exemptions in several countries, including tax holidays and special tax reserves to support export marketing costs

- diversification incentives -- the tying of export incentives to specific regions and the tying of taxation incentives to the quality of export performance and types of software products exported

- the more efficient use of multinational channels, beyond software mandates, to advance software training and development in Canada

- software development initiatives such as Singapore's International Consultancy Service Incentive and the Capital Pioneer Status Incentive, and the recent software development recommendations of Brazil's SEI

- specific AI incentives, policy options and development strategies to best appropriate the benefits of US and Japanese AI research, to stimulate Canadian research and product development in this area, and to transfer the benefits of this work to Canadian industry

- instruments for software funding and initiatives for channelling investment priorities into "technologically higher-level" software

- instruments to develop AI software in the universities

- government procurement instruments for selected software areas (such as computer-based training and learning)

- the granting of tax incentives and special credit lines for the production of new software applications and other measures.

1.4 Outline

Chapter two of this report presents basic information pertinent to the economics of the software industry and emphasizes the importance of "fourth generation" software.

Chapter three examines hardware developments of relevance to future software markets and to policy support measures, and chapter four describes software industrial and marketing trends. This discussion is deepened in chapter five with an examination of technical, industrial and marketing trends in software for microprocessors, microcomputers, industrial automation, and education and training.

Chapter six delineates the software industries and respective main support initiatives in five countries: Canada, Brazil, Singapore, the US and Japan. Chapter seven focuses on Canadian policy support for its software industry in five areas: tax/fiscal policy, R&D policy, procurement policy, educational/training policy, export policy and marketing policy. Within each of these areas, federal support measures are summarized and critiqued, and recommendations are made concerning how they might be improved.

Finally, a note on terminology. Throughout this work, personal computers are referred to as micros or, after the Japanese, as perscoms.

2.0 BACKGROUND

Computer software is normally classified into three or four categories according to its use -- systems software, applications software, data bases and applications generators.

Systems software regulates a computer's "metabolism", i.e. how data moves in and out and through the computer. Systems software may manage the running of several programs simultaneously, organizes data which is stored in memory and regulates access to data according to specific program needs. Systems software also schedules and prepares data for printing and hard copy, and includes "compilers", whose purpose is to transfer programs which have been written and stored in symbolic languages into a machine code for computer execution.

Applications software, on the other hand, is always used to tell a computer to perform a specific end user task.

Applications tools or programming aids involve software which enables a programmer to more easily write specific applications. Using an applications tool, the writer with comparatively low level writing skills may be able to describe on the computer terminal general properties of a software program he wants and then receive a partially written program.

In many classifications of software, applications tools are considered as a part of systems software, but actual programmer productivity is increasing so slowly in North America (approximately 4% annually) that a number of firms are emerging which specialize in such programming productivity tools.

Data bases are also sometimes considered to be software. Examples of computer data bases might include textual materials such as books or lists, visual materials such as pictures and films, and other information. The reason a data base is sometimes included as software is that many data bases do not comprise mere lists of information but also may contain specific programs of how to structure, update and how to retrieve data. This is especially true of so called intelligent data bases, those whose organization and retrieval are based on artificial intelligence notions. Software may also be packaged and sold "over-the-counter" or custom produced for a single customer. In the former case, no revenues are being received during development. Custom software firms must be located near the buyer firms, while location is comparatively unimportant for packaged software.

Marketing techniques, costs and distribution channels are also different for custom and packaged software, and the packaged industry is so new that a government tax structure more appropriate for traditional manufacturing has subjected this industry to devastating uncertainties.

Until 1969 when the US Department of Justice forced IBM to unbundle and separately sell software from hardware, software was virtually given away. However, the development of systems software now may cost hundreds of millions of dollars and may no longer be considered an ancillory to a computer.

According to Wescom estimates, over a computer's lifetime most owners will spend in excess of $2 on software for every $1 they spend on a computer. Within large corporations, users are still finding it very difficult to write their own applications, since software writing still is a time consuming and tedious process, and corporate users are learning that when they want to run new applications on their computers, they don't have enough money or programming time.

> "In fact, in many companies users must now wait as long as eighteen months for new programs to be written." (3)

Within the corporate world, considerable time and money may be cut by purchasing software packages which may be modified to a firm's specific needs. It is generally estimated that purchasing outside software programs instead of creating it in-house can reduce the development cost by 50%-75%. In fact, US corporations in 1983 increased purchases of software packages 50%, and sales of perscom business software packages increased 74% to exceed $1 billion.[4]

Since today's computer user is frequently a business person rather than a programmer who knows esoteric languages, how easily a program may be used is as important as what the end user actually gets.

2.1 Applications Tools

Applications generators, or software engineering aids, are increasingly used for semi-automatic programming. The idea is to provide a means of producing software programs simply by describing the requirements of the program. Here a user may generate a program by answering questions asked by an expert system, a program which uses explicitly represented knowledge and inference rules to solve specialized problems.

Another engineering aid involves modular software; components of large software packets are frequently reused across many programs, and in an average program, only 10% may be original work with the rest being modifications of common sub-routines. Modular software attempts to make available these frequently reused routines, and several software libraries in North America are applications-specific. Examples include McCormick and Dodges' AFS (for financial design) and IMSL's library of scientific software modules. Several critics have suggested that the development of these programming aids and applications generators may partially revitalize the custom software industry.

2.2 Fourth Generation Languages

Applications generators are themselves often written in "fourth generation" languages, based on concepts derived from artificial intelligence.

First generation languages, the machine language of 0's and 1's, were used exclusively by data processing professionals, while second generation languages, all of the assembler languages, also were used exclusively by DP professionals. Third generation languages such as Basic, Cobal and Fortran, also called "higher level" languages, made programming a computer easier and were utilized by a few end users in addition to DP professionals. Fourth generation languages made communicating with computers even easier and the next (fifth) generation currently being developed will involve direct voice recognition.

Writing an application in Basic or other third generation languages necessitates a strict adherence to tedious rules or procedures. Leave out a comma and the program does not work. People, however, talk in languages which are often ungrammatical and ambiguous, and the fourth and fifth generation (non-procedural) languages involve attempts to teach the computer to "understand" complex everyday English rather than have people learn computer languages.

Since it is generally estimated that the average knowledge worker spends about 70%-75% of his or her time in either searching for or manipulating information, current developments in fourth and fifth generation languages will for the first time truly automate office tasks and reduce search and manipulation time by up to one-third. Also, computers will not become common items like radios or telephones until they use natural languages, since most people won't learn the tedious third generation languages.

The main market force pervading all forms of software discussed in this work involved products incorporating, and being developed through, fourth generation languages. Univac has estimated that the world market for fourth generation software will be approximately $50 billion around 1994.

One fourth generation product of Cognos, Powerhouse, has already been installed on 12% of the more than 10,000 HP 3000s in the world. Indeed, the availability of one fourth generation package, Mapper of Univac, was the main reason the city of London, England bought the Univac Series 1100 computers (to get Mapper), which has also been bought by four southern Ontario cities -- Windsor, London, Mississauga and Scarborough. Windsor has:

> "... developed a building inspection system in about 1é weeks using Mapper which would normally have taken several months using conventional programming. London monitored development of the system, used it, and was so impressed that they bought their own system, including an 1100." (5)

Fourth generation software, in other words, is now a major selling factor for computers.

2.3 Software Frames

One new programming aid to ease writing applications software is "computer-aided programming" of Netron Inc. of Downsville, Ontario. Similar to computer-aided manufacturing, computer-aided programming utilizes the computer to handle the majority of the work necessary to build and maintain programming codes. Any custom code is automatically spliced in with the useable code, and such programming facilitates non-technical users to produce software business applications, which today can only be written by more experienced programmers. The Netron Inc. program partially automates the process of both writing and maintaining applications software and also allows customized programs to be more rapidly designed and assembled.

The basis of this new technology involves reusable software components which are called software frames, in contrast to software programs. (Programs assemble and manipulate data, while frames assemble and manipulate programs):

> "Unlike subroutines, as a frame evolves, it never compromises programs written in an earlier version of the frame. This universal upward compatability is what makes the frame approach to automatic programming so much more efficient than subroutines, the industry's first and most common approach to reuseable codes." (6)

In mid-1984 a joint venture was announced between Netron Corp. and Allied Canada Inc. to establish a Canadian centre for software engineering and artificial intelligence. In particular, this joint venture involves the implementation of Netron's computer-aided programming (CAP) technology with source code in Ada instead of Cobal. (Allied Canada Inc., based in Mississauga, Ontario, is the software division of Noma Industries Ltd.)

The productivity of CAP has been evaluated by Dr. Judith Drake.[7] Drake concludes that CAP users on a Wang VS can produce an average number of 2,000 lines of finished Cobal code daily. (Programmers normally produce around 20 lines of code daily.)

CAP relies heavily on AI technology developed at the Massachusetts Institute of Technology. Each "frame" may be thought of as an expert in constructing a specific type of program, such as data entry. Programmers enter into interactive dialogues in CAP and can choose from a large range of options to outline the basic features of the program they are developing. The completed code in Cobal or Ada is then automatically produced, with the programmers' specifications. Subsequent code can be added to the program or screen corrections can be made, and at the end of all additions and alterations, CAP automatically performs the customization.

2.4 Software Compatability

Any computer has the natural ability to understand instructions in only one language, its "native" machine language. This language differs from computer to computer. Compatibility problems occur when an application is moved from a computer with one operating system to another. Often, it must be entirely re-written.

Initially, "high level" languages, such as Fortran, were intended to be machine independent; each computer manufacturer would provide "translators" to transform the high level language into the native language of that machine. However, "dialects" of these languages developed over time, and machine-independence never resulted.

Although microcomputer operating systems may one day become truly standardized, presently incompatibility is a major problem, and even when a program is written in the same language, it can't be taken, for example, from an IBM PC to an Apple system. Manufacturers see a distinct advantage in maintaining these incompatibilities. Apple Computer Inc., for instance, enjoys an advantage in the educational world because there is a considerable amount of content available for Apples that will not run on other perscoms.

2.5 Embedded Software

A major component of software products by computer manufacturers involves firmware, or software which is placed permanently into read-only memory chips.

Although the state of the production technology somewhat limits the number of efficient firmware applications, VLSI advances will rapidly increase firmware applications over the next half decade. Virtually all major computer manufacturers are researching firmware, and when software is finally copyrighted in Canada, this will reduce competition from the IBM clone/compatible manufacturers of computers. This rebundling trend, as it is being called, is exemplified in IBM's 4321 computer with an embedded operating system.

Thus far, however, little applications software has been embedded, with the exception of perscom applications such as Apple's MacIntosh and Lisa computers, but one trend over the next three years will involve applications-specific 32 bit computers with embedded software, such as CAD/CAM computers.

Another example of embedded software is the increasing amount of software being written for handheld or lap computers. For example, in mid-1984 several technical software libraries containing 100 individual programs for Radio Shack TRS 80 PC IIs and the Sharp PC 1500 went onto the market. These libraries, containing general topics such as electrical engineering, finance, statistics/mathematics and business graphics, are all available in solid state modules which plug into the back of the handhelds.

2.6 The Transportability of Software

Over the next five years the transportability of software will increasingly free writers from being dependent on the individual computer vendor and will also allow end users to no longer be tied to a single piece of equipment. More importantly, software transportability will have radical effects on the rate of software production in the industry and the rate at which new software is available.

Also, the portability of software will slightly decrease the present manpower requirements for software writers in the advanced economies.

(By perusing recent editions of the software jobs-wanted ads in the New York Times and papers of other urban areas, one can get a good notion of the specific types of software skills presently required in North America. Applicants most in demand are those with experience in C, Unix, the design of graphics software and portability software.)

Software is said to be portable when it can be made to operate on a computer different from the one it was originally written on at less cost than entire rewriting. Many of the big US software houses have a "portable system" which serves as the basis for developing their applications software, and such systems are usually proprietary. As a result, most software in North American industry is not portable. One study has noted that:

> "When you prepare a (portable) software package for one computer, it will run on many others with little or no extra investment. Thus the original investment in software marketing, user support and other activities supports the product, which has a much larger potential market. Portability and its associated technical advantages should be disseminated through a central agency made available to software developers." (8)

There are a number of software tools, such as the University of Waterloo's WSL, to make portable software. Using the language of WSL called MIL, a program can be written to run on IBM, DEC, Intel and Motorola processors. Other truly portable languages include C, Port and Eh, also developed at the University of Waterloo. WSL is a software development system which supports any of the above languages.

The advantages of having portable software extend beyond the issue of how much code has to be rewritten for translation between machines and involves possible rewriting of all documentation and textual materials. In fact, if programming tools themselves are written in ways which are portable, real economic gains are involved.

The Computer Systems Group at the University of Waterloo developed WSL in 1980. Since then Waterloo has developed interpreters for Cobal, APL, Pascal, Fortran and Basic, which will run the same way on eight or nine computers, including mainframes and microcomputers. Packages have been developed on WSL for general ledger, small businesses, word processing and relational data bases.

2.7 Software Valuation

There are virtually no uniform ways of estimating the value or costs of software.

Revenue Canada has long treated software as a tangible asset and requires that all software development costs be capitalized, with depreciation of up to 100% being allowed.

Customs and Excise and Revenue Canada both consider that systems software be charged duty at the same rate as the hardware when it is included in the price of the hardware. Applications and systems software separately sold from hardware is not dutiable, and duty is charged merely on the media on which the software enters Canada. If one carries software into Canada on punched cards, for example, or shoots it over satellites, there will be no duty.

On the other hand, Revenue Canada requires that software always be treated as a tangible asset when software is licenced for export to eastern block countries.

It is clear then that software is sometimes treated as a tangible and at other times as an intangible by the government.

Keeping in mind the difference between cost and value, it is almost impossible to attach a market value to a software package unless it is sold, in which case the value may be considered the value of the revenues generated. Software packages, in fact, are often costed out in terms of what the market will bear and not in terms of factors such as development, marketing and maintenance costs, plus company profit. This factor explains some of the bizarre price movements of software which is sold to end users rather than leased. Also, if a poorly selling $30 package has its price raised to $3,000, it may sell better.

Although some unbundled software is still given away "free", it is generally assumed that costs of this free software were already included in the hardware price. In fact, it is commonly estimated that one-quarter of the cost of hardware can now be attributed to "free" software.

For all of these reasons, revenue received for a software package may not be an accurate measure of the package's value.

Other ways of valuing software involve replacement costs (the costs involved in replacing the software, which may change due to the availability of an off-the-shelf package and inflation) and the use value of the software (according to which programs are valued according to the importance of the function they perform within an organization).

2.8 User Antagonistic Software

The generally false claims of "user friendliness" become extremely important when one realizes a firm that spends a half million dollars for microcomputer software might have to spend three times that much training users. Most of the user friendly software works fine for people who are familiar with computers, but if one is a computer novice, that software is user-antagonistic.

Current "user friendly" developments such as windowing software, menus and applications which facilitate data swapping (all subsequently discussed) have been labelled as gimmicks and pacifiers. Many analysts think of these as transitional vehicles to true usable software which will be based on AI notions.

In fact, the most recent giant stride toward user-friendly software, we shall see, involves natural query languages. TI for example sells a natural language option which can be added to its professional computer so users may make queries of remote data bases in typed English; and with Salvo from Software Automation Inc.

of Dallas one can even misspell. Salvo functions as a DBMS plus natural language query and an applications generator. In addition, Intellect of the Artificial Intelligence Corp. now runs on IBM PCs, which may be connected to any mainframe Intellect installations, and standalone versions are expected by the end of 1984.

It is expected that English query languages will be common on the market by the end of 1985. Generally, the natural language firms think that other vendors combined use of menus, windows, mice, etc., are mere tricks to entice users.

The importance of ease of use and "user friendliness" is emphasized by some recent cost data on developing expert systems.

Dr. R.G. Smith[9] of Schlumberger/Doll has described the development cost of Dipmeter Advisor, an expert system for interpreting geological data. The code breakdown for production was as follows:

1. User Interface - 42% of total code.
2. Knowledge Base - 22% of total code.
3. Support - 15% of total code.
4. Feature Detection - 13% of total code.
5. Imprints - 8% of total code.

In other words, much of the development time of expert systems goes into user interfaces and not into the construction of knowledge bases, as is often reported. When an expert system has been commercialized, it is extremely important that it be easy for naive users, since there will not normally be AI consultants available to instruct end users.

2.9 Person-Machine Interface

Speech is the most natural form of human communication, and it is widely expected that within the decade, speech linkage with computers will come onto the market. However, in the meantime, one must use menus, mice, keyword searches, pictograms and various other techniques to search for information. As anyone knows who regularly uses these search techniques, tutorial commands and menus are extraneously and repeatedly presented, and this redundancy significantly slows down search time. Consequently, software designers are now trying to ensure that a computer user does not have to slog through repeated extra information when searching data bases. One recent solution in this area is a "time-delay activator" in the software. In other words, when users type in a command in less than, say, three seconds after being asked for the command, the software skips over a menu which lists options and assumes the user is competent.

In other areas of person-machine research, attempts are being made to reduce the number of language commands which any user must know to operate a computer. For example, in the MacIntosh and Lisa computers of Apple Computer Inc., typed commands have been replaced with a series of tiny pictures, e.g. a wastepaper basket to indicate discarding a file. To instruct a computer to eliminate this file from memory, the user simply moves the pointer to the part of the screen that contains the file and presses a single command.

There are also efforts to standardize commands, since many software packets now utilize different commands to do the same thing.

2.10 Software Production

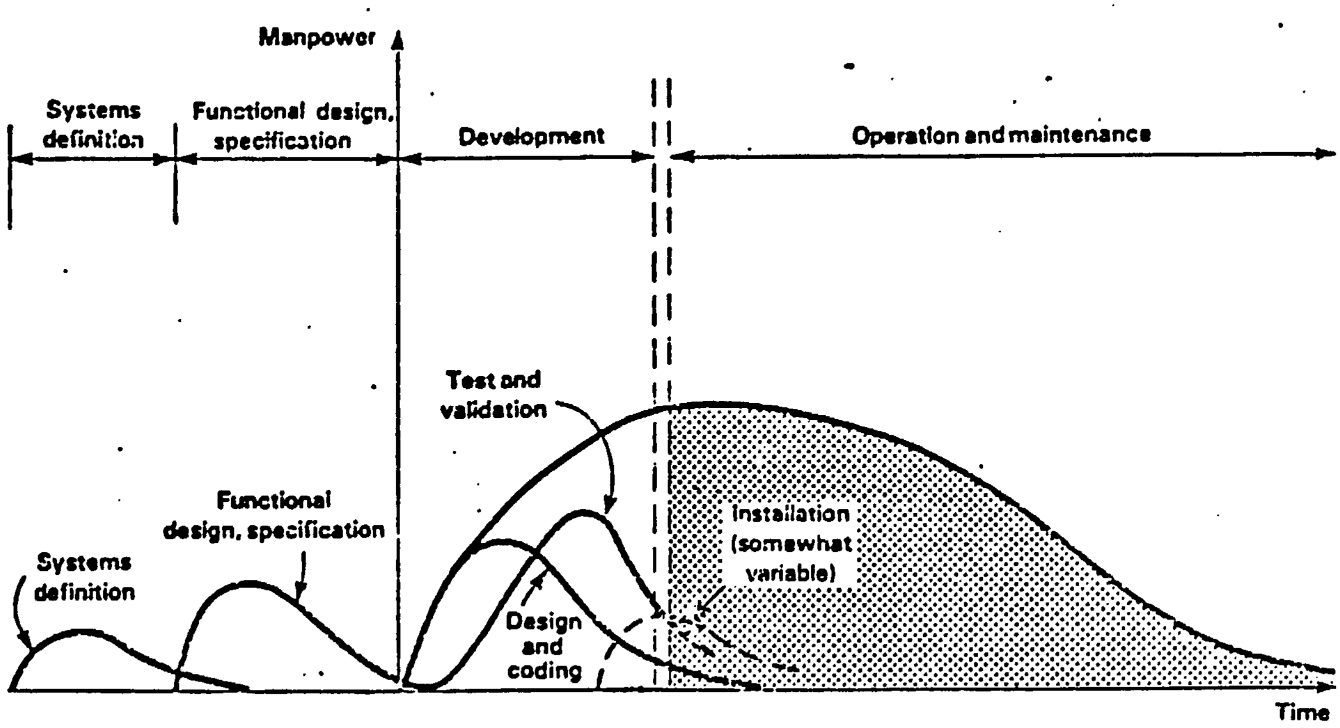
Traditional software production is often thought of as involving four steps:

1. needs analysis and design
2. specification
3. the actual programming
4. testing and debugging. (Exhibit 2-1)

In needs analysis the idea is to unambiguously logically represent the problem to be solved. For applications software, this preliminary stage of analysis must necessarily involve the end users of the software and with the needs analysis of systems software problems, systems users and specialists must be consulted.

EXHIBIT 2-1

The Software Life Cycle



Source: Computer Services Association

The second stage involves a detailed specification of both the input and output formats and the logical structure of the software.

In the third stage, actual programming, the software writer produces symbolic instructions which translate the logical procedures of the second stage into a language the computer can understand and execute. If a programmer writes symbolic instructions, with each instruction generating only one machine instruction, he is using "machine or assembler language". However, when the programmer's symbolic instructions generate several machine instructions, he is said to be using a "higher level programming language".

The most widely used higher level computer languages today are Cobol for business applications, Fortran for scientific applications, Pascal for scientific and management, Basic for simple engineering applications, and APL for interactive purposes. In several OECD countries in national surveys of software writers' languages, usually around half use Cobol, with around one-third using Fortran.

In spite of what has been written about the productivity changes resulting from the introduction of higher level languages, a large business program might typically include 150,000 lines, and a large scientific program a half million to a million lines of source instructions. One can understand why this stage of software production is so labour intensive, given the fact that a typical programmer can write about 20 source instructions per day.

Although there is a real range in the amount and quality of code a programmer can write per day, it makes little sense to apply productivity measures for this industry which are derived by analogy from manufacturing, i.e. output per worker hour becomes lines of code per worker hour. Productivity here depends on both quality, i.e. the content of the code, and quantity of production. With much software writing, however, it is really not possible to make a reliable cost or schedule estimate until the end of the design phase. But with many projects, so much money has already been put into design time that cost overruns are large, and sometimes at the end of the design phase there is a renegotiation which might eliminate a very costly function. Also, any error which is perceived at the design stage of software writing may be orders of magnitude less expensive to remove than one seen in the final production stage. Design requirements are also very heavily tied to error-free maintenance, since much programming time is taking up with maintaining previously written software.

The final stage involves software testing, debugging and maintenance. Productivity research on software writing, in fact, is focusing on both writing and maintaining software. Elimination of errors can be very expensive and can take a very long time. Errors may lie in three or four instructions amongst hundreds of thousands, and the number of errors in software programs in fact has been rising significantly during the 1970s and early 1980s, given the general decline in software writer qualifications, which itself arose because of the massive increase in the number of software writers.

In summary, software writing is enormously expensive, and even more expensive for systems software than applications, since systems programs can tolerate virtually no errors. Thus, if a single application package might cost typically $1-$2 million to develop, systems software may cost between 10 to 200 times the cost of the hardware it is written for; in other words, hundreds of millions of dollars. (Development of MVS, an IBM operating system, cost IBM thousands of millions of dollars.)

The economics of packaged software, in fact, shows many of the pricing characteristics in the neo-classical theory of the economics of information introduced in the mid 1970s, by economists such as Kenneth Arrow, Jack Hirschlifer and others.[10] Furthermore, given the enormous potential user-base for system software (everyone with a computer), this packaged approach has diffused more rapidly for systems software than for applications software.

Generally the spread in price between a software packet and its custom equivalent will depend on the potential number of users of the software. This spread may vary from 1:10 to 1:3000.

2.11 Venture Capital

Although Canadian venture capitalists are still quite leery of software and intangibles in general, in the US the venture capitalists have jumped into the software industry with a leap. Although four years ago less than 20 US software firms had attracted venture capital support in the US, this number has more than quadrupled by January 1984.

US software stock has also significantly increased in value over the past few years, performing 4.6 times as well as the Standard and Poor's Composite Index over a three year period.[11]

2.12 Maintenance

It is often claimed that approximately one-third of the costs of software over its product life cycle are involved in production and testing, and approximately two-thirds of these costs involve maintenance. (Other studies put the percentage of total software costs at 75% maintenance activities, with 25% for development.)

Another point to be noted about maintenance is this: since the industrial trend is away from custom and toward packaged software, end users usually cannot modify it, and maintenance obviously requires work by the package designer, especially if the program is supplied in the form of machine code rather than as a source program. The latter, in fact, is increasingly the case, we shall see, because of the issue of legal protection of software.

Although there is no consensus as to what precisely constitutes software maintenance, there are two common ways of estimating the effort which goes into maintenance, one involving annual operation costs, the other involving development costs.

Each of these methods in turn can be performed in several ways, e.g. with respect to the latter, estimations of maintenance costs may be expressed as a percentage of total development costs for the software or as a percentage of the total life cycle costs of the software, including initial development costs.

There are a number of problems, however, with this method of estimating maintenance percentage:

1. Since development costs are often expressed in actual dollars rather than current dollar figures, in inflationary times the cost of maintaining a software package may be higher, as expressed in real dollars, than its development cost.

   > "Furthermore, since inflation cannot be accurately predicted, unless the development costs are revalued to current dollars each year, it will not be possible to obtain a reliable and consistent development/maintenance cost ratio. A further problem is that the development cost may be unknown, e.g. a vendor-supplied operating system which may be a no-cost item may require regular and extensive maintenance activities." (12)

2. Maintenance is not costed over the life cycle of software but is concentrated in the first couple of years and then decreases.

3. It is difficult to devise figures indicating "average maintenance costs" with respect to development costs, since these two should vary with respect to each other; in other words, the more effort given to development, the less maintenance should be needed for packaged software.

4. There are <u>different types of maintenance functions</u>, some relating to factors such as size and complexity of the software package (i.e. routine debugging) and other functions relating to system generation (type changes); and these latter have little relationship to either the size or cost of the development phase.

As a percentage of total cost, development/maintenance cost ratios have varied widely, and this variation is at least partially accounted for by the different meanings of "maintenance". Maintenance, in fact, includes routine changes such as those in systems software, data and input files, debugging, etc., but it also includes enhancements, i.e. the addition of new features without the replacement of existing functions. However, such an enhancement might validly be thought of as further development activity and not merely maintenance. Lientz and Swanson[13] have examined applications maintenance and found that enhancements comprised 42% of all maintenance functions.[14] Some enhancements, however, might involve routine improving of existing code.

With respect to maintenance estimations as a percentage of annual operation costs, if maintenance is thought of as including all enhancements, then it is estimated in 1981 that software maintenance activity involved about 66% of all software resources, and development only about 34%.[15] However, excluding enhancements from maintenance, percentages are reversed and become development 66% of total software budget and maintenance only 34%.

2.13 Software/Hardware Cost Ratios

It is often claimed that software now comprises 50%-75% of total software/hardware costs of new products and manufacturing processes. However, it must be noted that most of these ratios prescribing hardware to software expenditures are made with varying degrees of analytical assumptions, and those persons arguing for the high software percentages usually include in software expenditures costs of programmers and computer specialists plus salaries for a claimed six additional persons who are necessary to support every ten software writers.

Secondly, hardware manufacturers' expenditures for software embedded in hardware is normally included in software costs in such estimates. This amount is typically claimed to be in the range of $1 to $3 spent for software for every hardware dollar.

In other words, such ratios consistently include indirect costs in software expenditures but do not include indirect costs in the expenditures for hardware. It may be more reasonable to think that there are a number of hardware/software cost ratio curves, each depending on the specific situation being examined.

2.14 Software Lifetime

A study by the Computer Communications Secretariat of the Department of Communications,[16] presenting a growth model for hardware and software in Canada, estimated a software lifetime of five to ten years. We have been able to find little data to support these figures. The lifetime of applications software might vary between six months and seven years, and between one and six years for operating systems, depending on the product.

Other factors relevant to determining the lifetime of a software program is when a package has existed for several years, the contemporary version, due to constant enhancements and maintenance, may little resemble the original. Also, higher level languages are now prolonging the lives of some applications software.

2.15 Exploratory Programming Environments and New Languages

Returning now to Exhibit 2-1 illustrating the software production stages, it must be noted that this scheme is applicable only for conventional programming techniques and <u>does not work</u> for large, complex programming problems.

It is extremely difficult to accurately specify parameters involved in a large software problem because the total complexity may preclude designers from anticipating all the ways their specific design rules will interact. The problem may also require continually changing requirements. But the large programming problem of changing specifications or requirements is almost impossible to solve utilizing conventional programming methodology. As one writer has noted:

> "Virtually all modern programming methodologies have predicated an assumption that a programming project is fundamentally a problem in implementation, rather than design. The design is supposed to be decided on first, based on specifications provided by the client; the implementation follows. This dichotomy is so important that it is

standard practice to recognize that a client may have only a partial understanding of his needs, so that extensive consultation may be required to ensure the complete specification of which the client will remain happy. This dialogue guarantees a fixed specification that will form a stable base for implementation. The vast bulk of existing programming technology, such as structured design methodology, is designed to ensure that the implementation does, in fact, follow the specification in a controlled fashion, rather than wandering off in some unpredictable direction." (17)

Today, large complex programming problems such as simulations are often not amenable to conventional techniques, since clients frequently do not know exactly what is required or what they want, and their statements about needs are often aspirations rather than problem specifications. Since most clients frequently have no programming experience whatsoever to guide such aspirations, a software writer must simply try a number of designs and see which of these work. After the writer's attempt to pin the client down, this specification is partitioned and defined into a structure reflecting this partitioning until the problem is divided into a large number of more easily solvable and implementable problems. The initial structure gives a certain control over the actual process of implementation as aided by a number of programming tools.

But with very large programming problems in which, for example, one adds a new piece of equipment totally different from the original equipment on which the specification is based, an entire new specification of the problem may have to be worked out, and conventional programming tools are of little use. The problem is that each change in the specification leads to further changes until the overall structure of the modules are decimated.

An alternative, however, to such a disaster is

"... not to abandon structured design for programming projects that are, or can be made to be, well defined ... instead we should recognize that some applications are best thought of as design problems rather than implementation projects. These problems require programming systems that allow the design to emerge from experimentation with the program, so that design and program develop together. Environments in which this is possible were first developed in artificial intelligence and computer graphics, two re search areas that are particularly prone to specification instability." (18)

Structured programming has actually been around since the beginning of the 1960s and involves an attempt to sever programs into relatively independent blocks which represent functions. The idea is to facilitate a division of labour in the programming, error detection and design phases of software production.

Specific computer languages were written around the idea of structured programming, such as Algol. Written mainly for scientific applications, Algol has not widely diffused to large numbers of computer programmers, but two of its progeny -- Ada and Pascal -- show more promise.

In 1977 a team from CII Honeywell-Bull was commissioned by the US Department of Defense to develop a superior structured language Ada. The American National Standards Institute and the EEC Commission gave their blessings to Ada in 1983, and the EEC Commission is backing Ada's distribution in Europe. The French are working hard to develop aids and applications for Ada's use in non-military applications, and Siemens-Bull-Alsys is developing Ada compilers, and a group comprising Christien Robsing, the Danish Datamatic Centre and Olivetti is developing engineering programming generators in Ada. The private sector also likes Ada, and several hardware/software manufacturers have prepared Ada compilers for a wide variety of computers (e.g. Supersoft's CPM compiler, Telesoft's Ada compiler for the IBM PC, and Western Digital's Ada compiler).

A number of critics, however, have argued that any attempt to make a universal computer language is ridiculous, because there are now too many computers to accommodate a single language. More importantly, a number of critics think that Ada will be immediately obsolete with the spread of new programming languages.

Pascal, developed in 1968 at the Eidgenossische Technische Hochschule of Zurich, has become the standard scientific computing language. It is only diffusing now to industry because this was the original language of many young computer programmers in their universities. Pascal, unfortunately, is not particularly portable, nor is the applications software written in Pascal.

These languages are all third generation languages. New approaches which are expected to supersede all of these languages involves logic programming and object-oriented programming. Using an artificial intelligence-based language called Prolog, invented at the Imperial College of London, the fifth generation computer project of Japan is developing a logical programming approach.

Some common properties of all of the new programming approaches, such as logic programming and object oriented programming, involve the absence of conventional structuring means based on redundancy. Secondly, these new approaches utilize "late binding", which allows a software writer to delay major writing commitments as long as possible. An example of late binding would be "dynamic variable timing", i.e. associating a specific data type with a variable name at the time the program runs rather than in the actual text of the written program. Another example would be the deferral of deciding value type until the program is run, because this allows a writer to experiment with the structure of the type itself. Another example includes the "dynamic binding of procedures". Beyond merely linking procedures at run time, the dynamic binding of procedures allows a writer to dynamically alter:

> "... the sub-procedures" invoked by a given piece of code, simply by changing the run time context. The simplest form of this is to allow procedures to be used as arguments or as the value variables. More sophisticated mechanisms allow procedure values to be computed or even encapsulated inside the data values on which they are to operate. This packaging of data and procedures into a single object, known as object oriented programming, is a very powerful technique." (19)

Logic programming, on the other hand, is based on logical descriptions of the nature and relations of the system which is being computerized. Another new approach, functional programming, involves "high level" functional instructions. Used extensively in the development of expert systems, one functional language is Lisp. Lisp is often defined as "a list of items which manipulates lists of items". Lisp programs, in other words, can themselves generate other programs.

Object oriented programming attempts to operate on objects, as defined by their relationships to other objects and their external structure. The first object oriented programming language, Simula, was developed in the Xerox Park Smalltalk project. Object oriented programming is expected to be very useful in representing systems whose external environment is important and which may be represented through graphic interfaces.

In summary, these new types of fourth generation programming will not achieve mere productivity gains in software writing but will involve the total restructuring of the several stage production process previously described, by semi-automatically generating complete programs from a description of the desired characteristics.

Finally, it must be noted that only a small percent of Canadian and US programmers are knowledgeable about these new programming approaches.

2.16 Artificial Intelligence

A software technology, then, central to Canadian industrial competitiveness, involves artificial intelligence-based approaches.

In a 1983 national conference on artificial intelligence held in August in Washington, more than 10 major computer firms were recruiting virtually anyone who knew anything about artificial intelligence. Given the scarcity of AI and cognitive scientists, salary demands are rising rapidly. As a result, universities are finding it increasingly difficult to compete with recent AI graduates in offering the salaries which industries can. For example, Schlumberger-Doll hired more than 50 AI scientists in 1983, a figure considerably larger than the total yearly AI output of PhD's in the United States. Also, since it is so valuable, most private sector AI research is now proprietary.

A benchmark of commercial acceptance of AI was IBM's June 1983 announcement that it had included natural language interfaces such as Intellect in its product line.

AI software enables a computer to emulate some aspects of human thought and is beginning to commercially come onto the market with products such as Intellect of Artificial Intelligence Corp., which allows naive users to request data from computers with typed English sentences rather than convoluted commands. Intellect is a non-procedural query language -- meaning that using it one does not have to follow the strict logical sequence of any third generation computer language. By the end of 1983, Intellect was working at more than 150 sites, and the firm currently has orders for hundreds of new installations.

Utilizing Intellect, a manager can query a corporate data base without having to use computer programmers to translate her queries into a code which the computer understands. In most businesses, management data bases are regularly utilized by less than five persons, usually computer experts. In most of the Intellect installations the typical number of users is 50-100, most of whom have never previously used a computer.

> "Intellect takes an English language request typed in by a user, translates it into a form that reflects the peculiarities of the data base, repeats this on the screen so the user can see the machine has understood correctly, and then summons up the information required ... the trick requires knowledge about the user's intentions and the background of his request. In Intellect, that knowledge is provided by interrogating the data base, often many times through the course of a single operation. Thus if asked 'What are the average earnings of a salesman in New York?', the program will check the data base to determine whether 'New York' is ambiguous or not ... but, if Intellect is asked 'How do the average earnings of salesmen in New York compare with those of salesmen in Chicago?', it will check the data base, learn that Chicago is a city, draw the conclusion that the New York referred to is also a city, and provide the desired comparison." (20)

Using non-procedural query languages such as Intellect, one does not have to construct a new knowledge base with each novel application, since this is done by the program automatically. However, with each new application one must construct a list of appropriate words, including company slang, which are used in this data base. This requires two to three weeks.

A major area then which artificial intelligence-based software has radically changed is how to search for information in computerized data and information bases.

Another natural language query system, Univision, includes split screen and windowing capabilities.

One natural query language for perscoms, Savvy of Excalibre Technologies, released in mid-1980, is a set of routines which analyze English statements and can perform functions against a data base. As opposed to the more common parsing methods, this product uses a pattern recognition technique. The advantage of Savvy over most interfaces is that Savvy users can actually misspell and omit words, and the system still runs. Of particular interest is the fact that when the product was first released, most AI and pattern recognition experts claimed that it was impossible, and this skepticism prevented Excalibre from getting any significant amount of venture capital in their early years of development. Although Savvy never took off on its initial vehicle, the Apple computer, in early 1982 a McDonald-Douglas minicomputer subsidiary, Microdata, advanced $500,000 to Excalibre as a royalty to be incurred by this firm's use of Savvy.[21]

Also commercially available are spread sheets which incorporate techniques from artificial intelligence. For example, Expert-ease, designed for the IBM PC, analyzes data and decisions for new computer users by learning from experience.

2.17 AI Market Trends

Although the commercial development of AI is just beginning now, with dozens of new start-up companies formed to exploit a wealth of existing publicly funded AI research during the last two decades in the US, new market trends are already apparent.

First of all, many of the start-up firms are assembling natural language interfaces together with expert systems for specialized purposes.

Secondly, "fourth generation" languages such as Smalltalk and Lisp are being used to prototype new AI work stations and many analysts predict that Lisp-based work stations will have up to 10-fold increases on programmer productivity.

Thirdly, early in the development of North American AI markets, software suppliers are attempting to lock in systems developers.[22] Only three companies -- Symbolics Inc., Lisp Machines Inc. and Xerox Corp. -- presently control a $50 million market for Lisp AI computers, but it is expected that IBM and the Japanese will soon enter this market.

Lisp, like all AI languages, is tedious and difficult to learn to use, and these new personal "symbolic computers" will make available the power of symbolic computing to end users without having to learn complicated AI languages.

The leading vendor of Lisp computers, Symbolics, is currently shipping about 30 a month and had their first profitable month in mid-1983. The Symbolics machine sells for around $32,000 and its main competitor, the Xerox Corp.'s Dandelion, is in the range of $90,000. Both are going after Apple's Lisa market. The third place contender, Lisp Machines Inc.'s Nu, sells for a little over $70,000 and also has Unix and Ethernet capabilities. Although this machine is being marketed as a development system, LMI is also hoping to attract customers who want a "delivery vehicle" -- a computer onto which AI turnkey applications are performed at end user installations.[23] (Within two years, competitors for the Lisp market will probably include Apollo Computer, Hewlitt-Packard, Digital Equipment and Sperry Corp.)

2.18 Canadian AI Activity

In Canada, although lack of support of AI research has led to a migration of our researchers to the US, several universities are active in theoretical aspects of artificial intelligence and are detailed in the Cognos study.[24]

Programming languages that use AI are being worked on at the National Research Council, the University of Western Ontario and Logo Computer Systems of Montreal, and natural language communication is being researched by the universities of Toronto, British Columbia and Alberta. R&D in machine translation is the focus of efforts of the University of Montreal and the University of Saskatchewan.

In the area of expert systems, research is being conducted at the Universities of Toronto, Saskatchewan and British Columbia.

The University of Waterloo is active in all of the areas mentioned so far, as is Canada's National Research Council and the Communications Research Centre at Shirley's Bay.

Although some AI related work is occurring in Canada's private sector, e.g. B.C.'s McDonald Detwiller & Assoc., GM Canada, Litton Industries and Diffracto of Windsor, Ontario, most of the AI research is occurring in universities.

AI-based techniques will critically influence virtually every type of software discussed in this work -- from industrial automation software to software for education -- and will also make major inroads in increasing programmer productivity. Although this work does not separately scrutinize AI software, the following sections review the main Cognos findings, and their policy recommendations are critiqued and supplemented in the last chapter.

2.19 Canadian AI Expertise

The 1984 Cognos study examined opportunities for Canada in the areas of artificial intelligence and specifically in machine translation and natural language processing. The study has noted that realistic assessments of opportunities must include, beyond discussions of technological advantage and market demand, factors such as:

> "... the existence of competition, the presence of industrial and commercial capability, the available of manpower and other resources, and the presence of institutional mechanisms capable of facilitating exploitation." (25)

The report summarizes that Canadian universities have a:

> "... fragmented and geographically dispersed scientific base in artificial intelligence; that the Canadian software industries are mainly branch plants of foreign owned multinationals; and that Canadian industry as yet has a (comparatively) low level of interest in AI technology." (26)

The study identifies Canadian AI research expertise in the areas of machine translation, vision systems for robots, computer learning, and expert systems in medical and resource industries. Canada also has a research expertise in the areas of person-machine interface and computational linguistics.

2.20 Canadian AI Firms

Canadian AI firms include Micronet Ltd. of Halifax, developing a self-learning controller for domestic hot water supplies; the Nexus Corp. of Ottawa, active in natural languages; Northwest Research Associates of Vancouver, started by one of the co-inventors of Prospector at SRI, has now implemented Prospector-II, which runs on an IBM PC plus a graphics tablet and aids in the formulation of models to locate ore bodies. New features include interactive capability for a map data base, and a query type of language. A full listing of Canadian AI companies is presented in Exhibit 2-2.

Given the necessity for rapid AI development in Canada, the study surveyed strategic industries such as forestry, finances and services, metal mining, energy and communications, to determine how sophisticated their current computer systems were, individual firms' awareness of AI techniques, specific AI application areas which might be developed and realistic time frames for market acceptance of AI products. A consideration in industry selection was significant levels of in-house R&D activities in these industries, which presumably might lead to Canadian applications development. Firms throughout these industries have a considerable level of awareness of AI techniques and a few firms have already begun to assess application areas for AI products.

EXHIBIT 2-2

Canadian AI Companies

Company Name	Area of Work
Nexa Corporation, Ottawa, Ontario	Investment Natural language
Micronet Limited, Halifax, Nova Scotia	Self-learning controller
Northwest Research Assoc. British Columbia	Expert systems (geological)
Cognicom Corp., Toronto, Ontario	Consulting in machine translation and natural language processing
Robotic Systems Int. Ltd., Sidney, B.C.	Robotics
Expert Systems Corp., Vancouver, B.C.	Expert systems Signal processing
N.W. Artificial Intelligence, Vancouver, B.C.	Expert systems (avalanche forecasting)
Gomi AI Systems, Kanata	Expert systems Medical

Source: Cognos

The main products cited as being potentially useful were natural query languages, expert systems and machine translation. Within resource industries such as fishing and agriculture, mining and forestry, the study also identified a specific application involving a combination of remote sensing/image analysis, which could play significant roles in monitoring resource industries and also noted that expert systems will be utilized to locate and manage resources. Most of the firms thought a realistic time frame for product implementation was five years to a decade away.

Although the Canadian AI capability is limited almost entirely to universities and funded at trivial levels compared to comparable projects in other countries, there is original work on knowledge representation, robot vision and image analysis, especially in connection with satellites, and machine translation.

> "Not only do we possess areas of special competence for which we have research programs already in place, but we also have certain problems of special national concern that lend themselves to AI methods ... such as aids for the management of the forestry industry, for the use of computer analysis of Landsat imagery or aids for medical research by computer-aided analysis of X-ray and other kinds of pictorial data. In addition ... there is the well known METEO system for translating meteorological reports ... there are also areas of application that are important in Canada which have not been pursued. For example, one of the most successful applications of expert systems AI work ... (has been) in the mechanization of expertise in prospecting for minerals and petroleum." (27)

2.21 AI Markets

2.21.1 Machine Translation

The study estimates that the Canadian, North American and world markets for translation are respectively \$100-\$190 million, \$400 million and \$4 billion (US) per year, respectively, and asks what percentage of this market would actually be accessible to various forms of machine translation, such as fully automated translation and human-aided machine translation. Although only 10% of the translation market could be addressed by fully automated machine translation, in the case of human/machine-aided translation, the percentage of the markets ranged from 20%-30%.

> "Machine translation ... is an area where Canada can compete on the international scene with relative ease; although there are products on the market and a few large scale efforts to develop systems in other countries, a long term commitment to R&D would be likely to give Canada an edge in MT and related technologies." (28)

2.21.2 Natural Language Processing

The next market area involves natural language processing software. Citing a DN Data Inc. August 1983 study, "The Emerging Artificial Intelligence Industry," the report predicts that natural language processing will capture almost one-quarter of the total market for software by 1990, and total revenues will be in excess of $1 billion US, just for North America. The main Canadian market driving factor identified involves aids to easily access computerized data banks and aids in the management of information. (Natural language processing in fact is an integral part of all national AI plans and has applications in many other areas, such as computer-aided training.)

2.21.3 Expert Systems

Expert systems were divided into five types: diagnostic systems, designer systems, advisor systems, tutor systems and inventive systems.[29] Although the US currently has more than 600 persons researching or commercially developing expert systems, last year Canada had one commercial expert systems project, and the 1983 DN Data study estimated that this market would grow by 55% annually, reaching $220 million by 1990.

2.21.4 Visual and Voice Recognition

The Cognos report also examined available market forecasts for optical character recognition, robotics and speech recognition, and describes Canadian interest in robotics, human-machine interface and computational vision and speech.

The North American visual and voice recognition markets, both growing at 50% annually, were respectively $30.3 million and $10.3 million in 1983. Software for visual recognition is the driving force now in robotics, but Canada has a small installed base of industrial robots and small estimated market growth. The DM Data Inc. study predicts 1990 markets for voice and visual recognition equipment/ software of over $1 billion.

Estimates were also made for remote sensing equipment and products which will increasingly incorporate AI-based computational vision software. It is thought that over the next decade and a half, the market will be between $1.1 and $2.8 billion US. Canada is, ironically, presently a world leader in computational vision applications for remote sensing and a late adopter of vision systems in robots.

2.21.5 AI Product Opportunities

The report suggested the following product development areas for Canada in the area of AI:

- computer-aided learning hardware and software

- image analysis systems for resource exploration and land surveillance

- machine translation

- expert systems in the fields of medical diagnosis, telecommunications and natural resource management

- intelligent control systems with the military.

With this background review, we now turn to an examination of hardware developments pertinent to software policy.

3.0 HARDWARE DEVELOPMENTS

Given the intricate connection between software and hardware, with an increased tendency to embede both applications and systems software onto ROM chips and incorporate them into computers, software policy may not be meaningfully examined apart from the entire field of computer and informatics developments. This chapter reviews some of the central developments in the area of computer hardware.

A basic change in the type of software being sold is due to the diffusion of personal computers. Software revenues, by machine type (mainframe, micros's, perscoms) are as follows. Software for micro's is now annually growing at 48% and 50% respectively in Canada and the US while the comparable figures for mini computer software and mainframe software are only 26%/39%, and 28%/31%.

3.1 Parallel Processing and Super Computers

Another change that will impact software production is the development of parallel processing computers. Roughly speaking, parallel processing computers operate by breaking up complex problems into many simple operations which can be simultaneously performed; each processor is dedicated or specialized to perform a single function, e.g. addition. In a crude sense this procedure is based on the way biologically intelligent organisms process information. (People and computers normally solve problems in entirely different ways. With simple addition or subtraction, human beings do not actually subtract or add the numbers but rather rely on a series of memorized sentences such as 2+3=5. People also utilize enormous parallelism in associated visual perceptions.)

A number of companies now are in the development stage of commercializing computers whose architecture and structure are based on biological notions. For example, Thinking Machines Corp. of Waltham, Massachusetts is developing "The Connection Machine", which will comprise several hundred thousand processors of small power and memory, with alterable links necessary to make different parallel architectures for changing applications.

Even more interesting are researchers who are trying to induce machines to deal in "hazy" concepts such as generalizing and making analogies. Also, researchers at Carnegie-Mellon and John Hopkins University are modelling computer architectures in which memory is distributed throughout the entire network. An advantage of "distributed networks" is the increase of redundancy. Human brains can experience extensive memory cell loss or damage without total losses of a specific item. The idea is to achieve this situation with computer memory.

Parallel processing computers divide up tasks amongst several comparatively cheap data processing devices instead of continuing the present trend of using faster single processors made with more exotic materials such as gallium arsenide. Over the next five years, Japan alone is spending more than $3 billion on parallel processing computers.

Even non-parallel processing (but fast) "super computers", as they are called, such as those of Cray Research Inc. of Minneapolis, have experienced major cost cuts and sales increases. The Cray computer, which a few years ago sold for $10 million, had its price cut to $5 million in 1983, and a number of industries such as aircraft and automobile manufacturers have replaced technology such as wind tunnels by simulations on these super computers. Many medical companies are using super computers to model drug effects, and some oil firms use super computers to model oil reservoirs and the movement of oil and liquids.

Canada, parenthetically, is not really taking advantage of this commercially available research tool. So-called super computers can, in one day of computing, accomplish what an earlier larger machine might take three or four years to do. The government presently has one super computer at Dalhousie University in Halifax, and a Cray super computer at the federal Atmospheric Environment Service in Dorval, Quebec. Plans for other super computers involve purchases for the Universities of Calgary and Alberta. In contrast, West Germany, France, Britain and Japan have more than 10 super computers each, and the US has 56. As a consequence, Canadian scientists have to use outmoded conventional computers rather than share time on a super computer.

Super computers such as Cray Research Inc.'s X/MP and Control Data Corporations's Cyber 205 are often used for simulating problems in science. The design of large scale integrated memory chips is virtually impossible without the use of such computers. (AT&T's Bell Laboratories used the Cray computer to design a 1 megabit memory chip.)

Europe is moving faster in this area than the US, and Germany's Science Council is planning to set up eight additional super computers in their universities, hooking together through network linkages Germany's 50 public universities. In 1983 the British also acquired several super computers at the University of London and Manchester, which have already been hooked into a network. In Japan, Hitachi Ltd. and Fujitsu have delivered their own models of super computers to the universities of Tokyo and Nagoya. The Japanese government is planning to heavily subsidize university purchases over the next decade of hundreds of domestically produced super computers for the universities.

In response to overseas developments, the National Science Foundation and the federally sponsored task force examining super computers have recommended a crash program for the US. (As part of this program, in 1983 $9 million was allocated to buy computer time for university scientists on existing US super computers.)

However, even these super fast non-parallel computers are quite slow for many industrial tasks which are emerging.

With the old "Von-Neumann" computers, as they are called, a central processing unit elicits data and program instructions in a sequential manner from memory. It then massages this data according to these instructions, shoots the data back to memory or performs further operations. Computation speed is limited by the specific speed of the processor's electronic circuitry, since only one processor is working. The speed of the circuit is itself determined by how rapidly transistors can turn on and off and how rapidly impulses flow through connections. It also depends on how hot computers get when they're operating, and non-parallel processing "super computers" are partially immersed in quantities of liquid coolants.

Many of the proposed technological solutions to making super computers faster -- such as Josephson Junctions -- have their technological faults and have been dropped.

As one writer has noted, ultimately this Von-Neumann bottleneck can only be overcome by switching to parallel processing.

> "The latent advantages of parallelism have long been recognized; even Von-Neumann was impressed by discoveries in the 1940s about how animal brains process information in a parallel manner. Had electronic hardware not been so costly in his day, he might have designed a machine more like a brain. That economic constraint no longer applies ... simply put many processors on a single chip. Processors could operate simultaneously on different parts of a problem, even specializing in performing particular operations at great speed. Another type of parallelism, called active or associated memory, can immediately speed up certain computer tasks, such as retrieving information from data bases by eliminating the role of central processor in searching through memory. Each data would be stored with its own processor, smart enough to respond to a centralized computer's calls ..." (30)

The changes in computer architecture will bring changes in software programming itself. With Von-Neumann computers a programmer has to spell out in the most exasperating detail and order everything he wants the computer to do. A number of analysts have asserted that the new parallel processing computers will make it much easier to program, since, for example, program steps will not have to be given in any specific order; but right now people don't know how to program parallel processing machines at all.

The central issue is this: how does one chop up a logical problem so that it can most efficiently run on multiple processors? Problems which necessarily must be solved sequentially, step after step, <u>cannot</u> be solved more rapidly by parallel processing computers, but many other types of problems can actually be severed into discreet portions and can be solved more rapidly by multiple processors.

3.2 Parallel Processing and New Computer Languages

Parallel processing computers currently being developed will enable one to execute multiple instructions simultaneously, but the power of such parallel processing computers depends on the birth of new computer languages.

There is an intricate connection between the architecture and the languages used on any computer. Normal, non-parallel computers or Von-Neumann computers, solve a problem, we saw, step by step in a sequential manner; the vast majority of computer languages in use today reflect this process.

One may think of computer languages as operating in a hierarchical way; at the most basic level there is a language called microcode which, embedded in the microchip, tells the switches to turn on and off in the correct order to carry out a specific calculation. One level up from microcode are specific computer languages used in operating systems such as Unix or MS/DOS. Operating systems "mediate" between the microcode and specific application programs, ensuring that each line of instruction in any application program and the data which is the subject of the applications program is directed to the right portion of the right chip at the right time in the right order.

The next stage in this hierarchical ordering are the various "higher level" languages such as Pascal and Basic, which are used for actually writing applications programs. These long listings of sequential steps must appear in a specific and strict order.

Errors incorporated into programs written this way are almost impossible to track down and very difficult to correct, because they are so tightly gnarled. If one changes one line, one may have to change 100 others lines in the program. Finally, these types of languages can tell the computer only how to do one thing at a time.

New, so-called "fourth generation languages" such as Lisp and Prolog avoid many of the above problems, and these languages break any large problem down into a series of short ones which can be worked on independently, and at the same time, by different chips in the computer. However, since there is a large existing investment by the advanced economies in the old sequential languages, it may be some time before new languages actually commercially replace these. The Economist estimates, for example, that the installed base of software written on IBM computers is approximately $300 billion.[31]

3.3 Light Signals and Fast Computers

Just as valves and vacuum tubes were replaced by transistors and integrated circuits, current design changes in optical switching devices are replacing electronic switching. This development is important in the context of super fast computers, since proton-based devices could operate extremely fast at room temperatures. (Both Josephson Junctions and transistors made of gallium arsenide must be chilled to very low temperatures.)

How to build a computer which uses light signals? One school of research holds that light should be used to shoot signals across chips and between them, but that on chips, light signals should be converted into electronic signals for conventional processing through transistors. Such hybrid chips would be made of gallium arsenide (which could respond to both light protons and electrons). Britain's Plessey has been doing research along this line, but such devices would be much slower than all-optical devices.

Another school of research is trying to produce entirely optical computing devices. Researchers at Heriot-Watt are using indium antimonide to manufacture an optical switch, which can be turned on or off in a millionth of a millionth of a second, or approximately a thousand times quicker than silicone based switches and several times faster than a gallium arsenide-based switch. Other new materials being tried out for optical switches involve lithium niobate and organics such as polydiacetylene. Xerox researchers are also looking into so-called "quasi-crystals", whose structure alters in the presence of an electric field.

Although all-optical computers are still many years away, optical switches have already been developed and used in inputing signals into fibre optics telecommunications systems.

Beams of light may be controlled with greater precision than beams of electrons, and an optical chip, unlike conventional electronic chips, may handle several different streams of light at the same time and use each stream for different applications.

> "In short, devices in an optical computer would have an intrinsic ability to do parallel processing ... and the architecture of the parallel processing optical computer could presumably be much simpler in basic design." (32)

But perhaps most importantly, the diffusion of optical computing devices might lead to basic changes in logics underlining current software languages. Electronic chips, in other words, have only two possible states: on or off, open or closed. Computers accordingly speak in a binary language whose logic or predicate calculus has two truth values: true and false. However, with optical computers, richer languages, more appropriate to artificial intelligence applications, could be, and are being, built up.

3.4 The Merging of Mini's and Perscoms

Many Canadian firms are writing software for minicomputers. After two years of continuous decline, the North American mini computer industry is improving. The industry leader, Digital Equipment Corp., actually had profits in the range of 20% in the first quarter of 1984, and other mini computer manufacturers show similar gains. However, this recovery may be temporary because of the growth of 32 bit perscoms, or low-priced desktop computers built on around a microprocessor chip. (Although most mini producers have tried to enter the perscom market, none is now a major contendor.)

Sales of perscoms in the US reached almost $10 billion in 1983, an increase of 85% from 1982. It is thought that sales will almost triple to $26 billion by 1987, according to International Data Corp.[33]

The first personal computers of Apple and Radio Shack on the market in 1977 were 8 bit machines. Several years later the IBM personal computer, twice as powerful, could process information in 16 bit pieces rather than in the 8 bit pieces of the previous generation. At the present time the next generation, 32 bit machines, are already coming onto the market. Having the processing power of a large mini-computer, these machines retail in the range of $4,000 to $15,000. Although they will first enter scientific/engineering markets, from there they will go directly into the office and other commercial applications. Such machines will be able to run natural language interfaces so that commands can be given to them in simple typed English rather than code. Increased processing power will also enable office workers to perform a number of jobs simultaneously, such as text processing, and mail on different parts of the screen. Applications such as speech generation, and limited expert systems which cannot occur on the 16 bit machines, will become practical. Roy Moffa of Digital Equipment Corp.,[34] which has recently produced the Microfax, a 32 bit perscom sees the new machines running artificial intelligence programs by 1989.

One of the main problems when the 16 bit machines were introduced was that they could not run software written for 8 bit machines. This error is not being made with the new generation of perscoms, and the new computers will be able to run on software which has been written for 16 bit perscoms.

Another result of the development of 32 bit perscoms is that over the next few years the traditional world markets for mini computers may significantly erode. This trend in turn will affect producers who write software exclusively for minis. Already over the past two years leading producers of mini-computers such as Data General have experienced decreased sales because perscoms are cutting into their low end products.

On the other hand, manufacturers of minis are working to compress their lines of "super mini" computers onto several chips, and as these super minis are reduced they will be able to use existing software which has already been written for the machines in non-compressed form.

It is widely predicted that the industry winners in this 32 bit round of perscoms will be IBM and American Telephone and Telegraph Co. Since the January 1, 1984 breakup, Western Electric Co., the manufacturing subsidiary of AT&T which previously made computers only for telephone companies, is freed to enter markets outside of telecommunications. AT&T's Bell Laboratories developed the Unix operating system, which has been extensively licenced to outside computer manufacturers. Unix seems to be emerging as a standard operating system for the 32 bit perscoms because it is excellent at facilitating several simultaneous tasks. It is thought that Western will be able to design computers which utilize Unix better than competing companies.

In fact, in March 1984 American Telephone and Telegraph announced the introduction of six new (personal and mini) computers. They also announced two new products to link computers together in office networks. These are the 3B Net which hooks together AT&T's 3B computers and the PC Interface, which allows any IBM-compatible perscoms to communicate with each other and with their new line of 3B computers. The 3B Net is also capable of interfacing with Ethernet (the local area network of Xerox). This new line of 3B computers is based on AT&T's own 32 bit microprocessor chip, and this microprocessor runs on AT&T's operating system, Unix. AT&T will be pushing hard to make Unix a standard for all perscoms.

Perhaps the most important factor in the rapid development of 32 bit perscoms is the fact that existing software has more or less exhausted the capabilities of existing 16 bit machines, such as the IBM PC. Integrated programs such as Symphony by Lotus Development Corp., Desq of Quarterdeck Office Systems and Framework of Ashton-Tate allow users to simultaneously perform a variety of tasks which hitherto were in separate programs -- such as word processing and spread sheets. Many of these new programs test the limits of the IBM PC and the Apple II. These limits involve the speed of the microprocessor (usually an Intel 8088); the limits of addressable memory; and the limits of graphic capability of any perscom's monitor.

3.5 Canadian Integrated Circuit Options

Given trends such as firmware, the incorporation of software into the circuitry of chips, should Canada continue to import most of her integrated circuits or should some be manufactured domestically, and if so what kind?

Although Mitel is currently developing, through reverse engineering, a proprietary microprocessor, there are a number of obvious reasons mitigating against extensive development of microprocessors. Development costs and times are enormous and they have to be sold in the millions to recoup costs. This industry is supported by a few large US and Japanese manufacturers.

Microprocessor chips, of course, have to be programmed, but each time the computing power of microprocessors doubles, the applications costs have gone up astronomically. It has, for example, been estimated that the average cost of developing a single application for a 16 bit microprocessor is in the range of $3-$5 million.[35] This is because a customer must design all the support chips (memory and input and output chips) and the computer board containing the microprocessor, and most customers simply do not have the sophistication requisite to place microprocessors into end products. This process also involves designing how the microprocessors will interact with input/output or memory chips, the system architecture plus the actual software, which is hard-wired into the architecture.

Custom chips, as distinguished from microprocessors, are tailor made for specific applications for end users who need them in small quantities. A custom chip usually does not have to be programmed at all. Since the marketing strategy of the big IC producers is to make a large number of standard chips and to force down unit costs, in general they don't like to produce custom chips.

Similarly, investments in random access memory chips are massive, and the production technology is very expensive and continuously changing.

The production of custom and semi-custom chips then looks more promising.

Customized circuits are widely utilized in read-only memories to convey embedded instructions to microprocessors in products ranging from automobiles to appliances and communications equipment. Dataquest has predicted that the total market for customized and semi-customized chips will grow from $725 million in 1981 to more than $3 billion by 1985, with most of this being in customized chips.[36] Several analysts, in fact, predict that by 1990 customized chips will comprise more than half of the total market for all integrated circuits.

It must be noted, however, that the production of custom chips is still very expensive and in the past has been often restricted to instances where the manufacturer could anticipate a large production run. It still may require between 50 and 100 person years to design a single custom chip which is free of faults, and the cost of computerized work stations used to design custom chips is still very high.

Canada presently has a limited number major producers of LSI chips -- including Mitel, Northern Telecom and Microtel Pacific Research. In addition to the above three who actually design their own circuits, other design houses are Mosaid, Siltronics and CAD/CAM Graphics Systems.

Several analysts have recommended that Canada set up production facilities for custom chips, and as of January 1984, the Alberta government is planning to set up a custom microchip fabrication plant. The government commissioned Bell Northern Research to conduct a study which concluded that Alberta must rapidly diversify its economy from its purely resource base into more technological aspects of this resource base, such as oil and gas communications, agriculture and medicine. The new facility would focus on niche's such as custom chips for remote sensing, farm equipment and pipeline control. Several other provinces are considering similar options. In the meantime, Canada's captive chip producers continue to advance. For example, in mid-1984 a $14.4 million joint venture was announced between BNR Ltd. and the National Research Council to develop gallium arsenide semi-conductors for use in high-speed optical communications and digital memories.

With this hardware briefing, the following sections examine general industrial and marketing trends in computer software.

4.0 SOFTWARE INDUSTRIAL AND MARKETING TRENDS

4.1 Software Houses as a Domestic Supply Source

Although the computer services industry comprises a small share of total software produced in OECD countries (compared to in-house production and other sources), it is the most rapidly growing component of the software industry. The following exhibits present data about the growth, composition and size of the computer services industry in several countries. (In 1982, for example, the US had over 6,000 software houses, with a total billing of $26.5 billion. Software houses are the main software supply source for small and medium sized firms in many OECD countries and in several countries, and not merely in Canada, a typically sized software house will comprise less than 10 persons and will have a limited geographical market size. Software houses, again in several countries and not merely Canada, have an uneven geographic distribution since they must settle in a comparatively dynamic market environment with a ready supply of user firms and potential customers. In the US, for example, software writers are concentrated on the west coast and in Canada in the Toronto-Ottawa-Montreal area.

4.2 Software Houses as an International Source

Increasingly, software houses are entering the international market. In spite of the fact that software design and maintenance requires a close physical interaction between buyer and seller, a number of software houses have been aggressively pursuing markets in foreign countries, often in association with national firms or via subsidiaries. For example, both DMR and Systemhouse have been active in Singapore. Given the restraints of software production software foreign subsidiaries may not function merely as marketing units as is often the case with manufacturing, but must also have the knowledge to engage in software maintenance and conversion at the very minimum, if not software design. One analyst has suggested that software international joint ventures will develop:

> "... where production takes place in developing countries and service and maintenance in developed ones due to relatively lower high skill labour costs and the fact that most of the market is in the advanced countries." (37)

EXHIBIT 4-1

Growth of Computer Services Market in the US in Comparison to Other Industrial Sectors (1976-1981)

Industry	Average Annual Growth Rate (%)
Computer Services	20
Securities	18
Banking	17
Electrical & Electronics	12
Total Manufacturing	11

Source: Input 1983.

EXHIBIT 4-2

CSI in Europe, 1980-1985 ($ millions)

Market Category	1980	1981	1985	Estimated Annual Growth (%)
Batch Processing	2,800	2,900	3,400	4.0
Remote Processing	1,900	2,200	3,800	15.2
Consulting Syst Implem	1,800	2,100	3,800	16.3
Software Products	900	1,100	2,600	22.9
Turnkey Systems	1,100	1,300	2,600	18.9
Total	2,500	9,600	16,200	13.14

Source: OECD.

EXHIBIT 4-3

Services Revenue Growth by Country

1981 % GDP Deflator	Country	1981 Revenues	% Growth	Processing Services	% Growth	Prof Services	% Growth
5.3	Austria	149	11.1	85	4.9	64	10.8
6.0	Belgium	397	13.4	210	9.9	187	17.6
8.8	Denmark	402	9.8	292	6.1	110	20.9
11.3	Finland	262	13.4	155	9.2	107	20.2
12.0	France	2,260	17.5	1,180	13.5	1,080	22.3
4.0	Germany	1,448	14.5	503	6.6	945	18.9
17.8	Italy	1,106	18.3	606	14.3	500	23.5
5.8	Netherlands	664	15.1	338	11.2	326	19.4
12.0	Norway	304	13.6	210	10.5	91	20.5
18.0	Portugal and						
18.8	Spain	315	18.9	193	16.2	122	23.2
10.0	Sweden	503	11.5	343	9.2	160	16.8
12.3	Switzerland	334	8.4	249	5.6	85	16.4
13.0	United Kingdom and Ireland	1,026	13.6	841	9.9	787	17.8
	Total	9,772	14.9	5,205	10.7	4,567	29.3

Source: Quantum Science/ECSA.

EXHIBIT 4-4

Growth of Data processing Expenditures in the United States, 1979-1985

	1979		1985	
	Millions	%	Millions	%
Software & services	7,850	15	21,050	19
Personnel	23,820	46	52,300	46
Hardware	16,510	32	31,700	28
Other	3,740	7	8,200	7
Total	51,920	100	113,250	100

Source: IDC

In this arrangement developing countries, of course, miss out on a major source of software revenue, maintenance.

Exhibit 4-5 presents a five year forecast for Canada's information processing industry, culminating in a $15.77 billion market by 1988, with software accounting for $4.62 billion. The 1983 revenues increased 18.3% over 1982, with software revenues increasing at 29%, in a recessionary period.

Although a number of North American software houses -- such as Geisco, MSA, ADR, Cincom, Cullinet, Pansophic and ADP -- all have invaded foreign markets, detailed statistics on the international rate of diffusion of software houses are scant, as this process is only occurring now. Also, international diffusion might take place through diverse channels, ranging from licensing arrangements, direct exporting, the setting up of subsidiaries through an affiliate of a parent firm, or through a joint venture.

Language Problems

One reason that systems software is diffusing more rapidly than applications software internationally is because first of all, systems software is usually bundled into the hardware which is sold and secondly, documentation is not as much of a problem as it is with applications software, since the computer experts who will be using the systems programs in the vast majority of cases already use English as a working language.

Documentation for applications software may, of course, be translated, but we have seen the results of such efforts in the documentation of the initial Japanese "perscom invasion" of North America in 1982. Also, the translation of software documentation is very expensive. For a moderately sized data base it might typically cost $1 million to translate the corresponding data base management system into another language.

EXHIBIT 4-5

Canadian Information Processing Industry Revenue Forecast to 1988
(C$ Million - Includes Exports)

	Actual			Forecast					
	1980	1981	1982	1983	1984	1985	1986	1987	1988
Hardware									
Sales, Lease, Rental	2,183	2,770	3,415	4,030	4,830	5,785	6,860	7,925	9,400
Maintenance	450	600	730	860	990	1,140	1,310	1,485	1,750
Total Hardware	2,633	3,370	4,145	4,890	5,820	6,925	8,170	9,410	11,150
Annual Growth (%)	25	28	23	18	19	19	18	15	18
Services									
Software									
Application Packages	62	95	154	220	310	410	560	725	1,010
Systems Packages	111	164	221	290	370	475	620	780	1,010
System Development	172	216	265	315	380	500	600	710	820
Total Software	345	475	640	825	1,060	1,385	1,780	2,215	2,840
Annual Growth (%)	50	38	35	29	28	31	29	24	28
Consulting, Education, Misc.	210	270	310	380	450	525	620	710	840
Annual Growth (%)	24	29	15	23	18	17	18	15	18
Processing Services	550	655	700	755	800	840	880	910	940
Annual Growth (%)	17	19	7	8	6	5	5	3	3
Total Services	1,105	1,400	1,650	1,960	2,310	2,750	3,280	3,835	4,620
Annual Growth	27	27	18	19	18	19	19	17	20
TOTAL REVENUES	3,738	4,770	5,795	6,850	8,130	9,675	11,450	13,245	15,770
ANNUAL GROWTH (%)	25	28	21	18	19	19	18	16	19

Source: Forecast by Evans Research Corporation based on (1) Statistics Canada figures for 1980 and 1981; (2) Evans Research Corporation's annual survey of the top firms in the Canadian computer industry.

p. 53

Increasingly, all countries' software houses are producing packages rather than custom software. The economics of the transition from custom to package production for Canadian firms has been discussed in Wills,[38] but suffice it to summarize here that package development costs are enormous, and financing of packages is totally different from that of custom software. Secondly, the marketing costs of packages have gone all out of whack compared to marketing costs for conventional manufacturing activities. The Lotus 1-2-3 software package cost several million dollars in marketing, and software houses which may not spend such large amounts on marketing often have to use intermediary distributors -- such as hardware manufacturers themselves, publishers of various sorts, software dealers, and so forth, and several independent software houses are now beginning to function as publishers for their own software and that of smaller firms. (It is often estimated after marketing costs that the original software designer's percent of revenues from a package is 20% or less of the final price.)

The main instrument in the international diffusion of computer software, until now, as with hardware, has been the multinational corporation. Between 1962 and 1976, in South Korea, for example, the Japanese multinationals were the source of 66% of 737 licensing agreements for microelectronics technology.[39] American multinationals, as is well known, were sources of much original informatics hardware and software in both Japan and South Korea. Also, Taiwan's electronics industries were founded on transfers of multinationals -- whose purpose was often to gain market access in such countries, and these trends are now being exhibited in Singapore's new optics industry.

Similar trends are occurring in computer software, and naturally enough it is mainly systems software (embedded into computers) which is being first diffused rather than applications software, which is often subject to cultural idiosyncracies and language problems.

Given the intricate connections between hardware architecture and software, an increased commitment on the part of computer manufacturers to software is resulting in a greater differentiation in market-available hardware. Also increasingly, the point of differentiation is not a physical property of the computer but the quality of system software.

Finally, as one study has noted:

> "It must be observed ... that manufacturers' recent hardware strategies seem likely to restore, in the not too distance future, de facto tied sale situations in several segments of the software market. More than through firmware, which will influence markets in a more remote future, it is through policies of not disseminating technological information about new hardware generations that the manufacturers have gathered to re-establish their control over software markets, because in this way they can be the only source of software when introducing new hardware models. In view of the very long lead time for software development, especially for systems software, independent suppliers may find it impossible to compete, at the critical moment when such markets are launched. The problem becomes particularly acute with the relinquishment of 'upward compatability' policies for software, eliminating the 'automatic' portability of existing independent software on new hardware." (40)

In other words, the introduction of the above policies would involve a return to a monopoly situation of tying in selected software markets.

4.3 Distributed Processing

Until the early 1980s, in-house supply by data processing departments in large businesses and government was the main source and also the main consumer of software.

However, the diffusion of perscoms into data processing departments of large corporations has altered this trend. Several 1982 surveys have shown that less than 20% of perscom software in large US corporations was actually produced by these departments, as compared with about 30% for user departments and the rest from outside sources, and that data processing departments were then still producing their software mainly for mini computers. Given the economics of packaged software, many DP departments are converting over to this source. Although in

1982 US firms spent around 10% of total computer expenditures on software from external sources, in 1978 they spent only 2%.[41]

Hardware manufacturers are also proposing new architecture strategies basically different from the centralized architectures of the 1960s and 1970s. Dataquest has predicted that by 1985 more than half of the installed computer bases in North America will be part of a computer network, and this change to distributed computing often means extensive revision of the existing software stock:

> "For instance, IBM through its new 4300 and 38 systems has been trying to generalize its new SNA distributed architecture, its new MBS operating system (and its top of the range version MBS/XA) and its new data base management systems MIS and DBII to be compatible with earlier software generations." (42)

Other companies such as ICL and C-II Honeywell-Bull are making similar transitions to distributed architectures, and as a result, within a single manufacturer's computer stock, there are now quite incompatible software programs with all the conversion problems involved in transfers, data bases, etc.

Another factor in distributed processing involves the growth of local area networks which can transmit data, images and voice, and allow the hooking together of integrated office equipment and linkage of perscoms with mainframes. Local area networks in both business and factories have an extremely high software component because they often necessitate that data format and applications software must run on several brands of microcomputers.

4.4 Consolidations and Mergers

There is now an increase in consolidations, mergers and acquisitions in the North American software industry. Those companies which are not involved in acquisitions and mergers are often attempting to go public and remain independent (Lotus, Pansophic, MSA, etc.), but only firms with real market presence can finance equity issues, and at least two Canadian software firms which recently have attempted to go public have not been able to. In the meantime, the banks are remaining intransient in their ignorance toward intangibles, and the other source of funding for new software companies, venture capitalists in Canada, have been so "prudent" that several software firms have been funded by venture capitalists in the United States.

Another problem with the development of packaged software is that long term funding is needed at the beginning of the project.

A number of analysts have noted that in virtually all countries besides the United States, the lack of a really developed venture capital market is a major hindrance in future growth of the microcomputer packaged software sector, the fastest growing component of all the computer and software industries.

Finally, given the closer weaving of systems and applications software (under the ruberic of "fourth generation" software), a number of houses (Cullinet, Cincom, etc.) chose a strategy of seeking complementary specialization in mergers with other software houses. Besides mergers, they have also entered into technical/ marketing co-operative agreements, the object being to share marketing and technical abilities. For example, in co-operative agreements to develop integrated software, the following firms have teamed up -- Visicorp with Informatics, Cullinet with Information Science, Martin Marietta with Mathematica, ADR with Visicorp, etc.

In the early days of the software industry then there was one group of firms which wrote system software and one group which concentrated on applications. This division is disappearing, and all of the major US systems software houses have now diversified into applications software. The application houses are responding by decreasing applications which work with several brands of system software and are increasingly writing for one system package.

The system houses are being forced to diversify, since the applications market is growing so much quicker than the systems market. In 1982 the US market for systems software comprised $2.6 billion, is growing at approximately 30%-35% annually, and will reach $11.7 billion by 1987. However, applications software, it is expected, will grow by more than 40% annually and will comprise more than $18.8 billion by 1988, up from $2.5 billion in 1982.[43] Also, big manufacturers such as IBM are trying to embede even more software and hire independent software suppliers. For example, in early 1983 IBM embedded a data base management system into their System 38 computer.

There are a number of ways for systems software firms to enter an application market; for example, engaging firms already in that field. (Cullinet has been helping Information Science Inc. to redesign payroll software.) Other systems software houses are simply buying applications firms. For example, in 1982 Computer Associates International Inc. of New York, a systems software house, acquired Capex Corp., an applications company. Other systems software firms are producing original applications packages. Cincom, for example, spent $20 million in 1982-83 to develop a manufacturing control package.

One problem here is that the markets for business applications and systems software are entirely different, and selling business packages to end users involves less technical knowledge than selling systems software to data processing personnel.

Other firms have started joint venture partnerships with applications firms. For example, Software AG has developed a data base management system with a program which hooks the data base management system together to applications packages and has been working with a number of applications firms to write new tasks which could be used with this program. These applications firms then sell packages together with a specific version of the AG component, with Software AG receiving a royalty for each version sold.

4.5 Changes in Industrial Structure

Until the past eighteen months or so, most North American software houses have sought out particular market niches such as systems, utility or applications software. (Stratification was true of software producers for either mainframe, minicomputers or perscoms.) System software was originally the prerogative of hardware manufacturers, with a number of smaller independent software houses focusing on applications. These distinctions are blurring.

There is a real consolidation trend in the industry, and in November 1983 IBM started to rapidly produce more applications programs. There is also a trend of acquisitions. Big computer manufacturers such as Burroughs, Prime Computer and Hewett-Packard are acquiring application software houses, and large manufacturers not involved in direct acquisitions, such as Digital, Sperry and Honeywell, have set up joint ventures with smaller software firms.[44]

At the same time, a number of software houses which produce software specifically for big mainframe computers, such as Cullinet, CAI and MSA, are acquiring perscom software firms. Also, firms such as Microsoft, which specialize in operating systems for perscoms, are rapidly adding applications programs to product lines. Finally, although in 1983 very little software was actually sold in book stores, a number of publishers and communication firms, such as Dow Jones, McGraw-Hill, Simon and Schuster, have been licensing software, which they plan to sell through their own distribution network.

4.6 Software Growth

As personal computers become increasingly a standardized off-the-shelf product like telephones and contain even more computing power, there is a continuous and severe shortage of software which can take advantage of these more powerful machines. As some notion of software growth, in 1981 North American software sales were less than $3 billion, but by the end of this year they are expected to be greater than $10 billion. Collectively in the US and Canada there are more than 4,000 software firms, and a number of analysts expect the software market to continue growing by 32% annually and thus exceeding $30 billion by 1988.[45]

Although today software revenues are equal to approximately 25%-28% of all hardware sold, if the above figure is correct, by 1988 software annual sales will comprise half of the hardware business.

The fastest growing portion of the software industry over the next three years will occur in software for perscoms and for super perscoms (collapsed minis), and estimates for annual growth in these areas have ranged from 30%-50%.

4.7 Marketing Costs

Software marketing has become even more expensive, especially in the packaged field. For example we saw the Lotus Development Corp., which produces the very successful 1-2-3 package, spent in excess of $1 million in a three month period of 1983 to launch 1-2-3. Advertising expenses are so great now that they are one of the main barriers for new entrants, and other US firms have spent in excess of $8 million to launch a single product.

But big software advertising budgets are not limited to companies writing mainly for perscoms but include those writing for mainframes and mini computers. Since minis have a restricted audience, firms do not have to put as much money into advertising. Nevertheless, several US firms such as Supply Data Research Inc., which write for these types of computers, significantly increased their advertising budget in 1984. The main difference is that with the supplanting of mainframes and minis by perscoms, the buyers of the software products are no longer in data processing departments of big organizations, but are end users. Consequently, not merely technological capabilities but benefits must be sold.

With increased competition, customer support and service becomes more important. For example, Micropro International Corp. recently invested several million dollars into the development of computer-based instruction for the use of its software, including WordStar.

Broadening a software product line also aids a firm in recouping the frequently larger cash amounts spent on advertising brand names. For example, Visicorp, which produced the widely successful VisiCalc program, is presently extending its software with an entire family of Vision software. As a result of this trend toward more diversified software products incorporated into a single package, many small companies will have to more specifically target market niches. A spokesperson for

Management Science of America has predicted that ultimately, "There will be several very large software companies, but there will be literally hundreds of smaller companies, with under $100 million in sales, that have specialized software niches." [46]

4.8 Summary

A basic structural change in the software industry is the blending of differences between types of software suppliers. Hitherto, software firms could be meaningfully divided into those that write systems software and those that write applications programs. This distinction no longer holds, and firms which have written software for mainframes are now entering the perscom software market, and firms which have hitherto concentrated on perscom software have teamed up with mainframe software producers. This is a natural enough development, given the trend in linking perscoms into company's mainframes. (Visicorp, for example, has joined up with Informatics General Corp. to produce VisiAnswer, a software package which allows users of an IBM PC to exchange data with an IBM mainframe.)

Given the increased competition and industrial mergers, one short-term trend involves a reduction in prices of software. As just one example, Visicorp has recently decreased the price of its Vision package from $495 to $95. Unfortunately at the same time as software prices will be falling, the development and marketing costs for new software will be increasing.

Having reviewed some central software industrial and marketing trends, in the next chapter we deepen this discussion through an examination of four types of software —for microprocessors, for microcomputers, for industrial automation and for computer-based learning and training.

5.0 FOUR TYPES OF SOFTWARE

5.1 Microprocessor Software

When microelectronics chips are placed into products and production processes, the first stage involves an analysis of the system by a microelectronics "specialist", who produces a detailed chart indicating the system architecture of the application. Given this architecture, the most efficient and cheap hardware configuration is designed around it.

Here there are two possible choices. The product may be comprised of off-the-shelf microelectronic components such as memory and microprocessor chips, or of custom chips (which in general do not have to be programmed at all). In the first case hardware costs are kept at a minimum, but software costs may spiral, while in the second case hardware costs may be high. However a number of factors -- such as the production of custom chips from standardized circuits or gate arrays (the construction of microprocessors using prefabricated microelectronics modules), the use of off-the-shelf software modules and actual design methods for customized circuits -- have all reduced costs. Even so, costs are still high compared to use of standardized circuits. A single custom chip, for example, might cost $200 as compared to $3 to $5 for a 64K RAM or $25 to $30 for a standard 16 bit microprocessor chip.

Several analysts have suggested that production of microprocessor software will be automated within the next decade and that expert systems will be able to analyze applications, design the system architecture and produce fully written software programs.

The Japanese had moved most quickly to incorporate microprocessors in consumer products, and the application rate of microprocessors in home appliances is shown in Exhibit 5-1.

EXHIBIT 5-1

Microprocessor Application Rate in Japanese Home Appliances

Appliance		*Year* 1977	1978	1979	1980	1982
VTR	Production volume (1000)	962	1559	2245	2837	4131
	Microcomputer applied unit (1000)	96	234	920	1560	3016
	%	10	15	41	55	73
Microwave oven	Production volume (1000)	1724	1857	1767	1577	1615
	Microcomputer applied unit (1000)	172	464	530	639	888
	%	10	25	30	40	55
Air conditioner	Production volume (1000)	2934	3864	5478	5100	5560
	Microcomputer applied unit (1000)	30	190	820	1275	1945
	%	1	5	15	25	35
Tuner	Production volume (1000)	2280	2680	2820	2950	3220
	Microcomputer applied unit (1000)		50	140	200	550
	%		2	5	8	17

Source: Report on Microcomputers, published by Japan's Electronic Industrial Promotion Association in March, 1980.

5.1.1 Code Writing and Testing

The main costs of placing chips into products, however, involve software design, writing and testing. In this case, the programming and testing involves either standard or custom circuits, and these require different skills and facilities.

Standard chips are programmed in widely available higher level languages. Unfortunately, a high level language program or source program which is compiled or translated into a machine program before occupying circuitry of a microprocessor takes up a lot of space in a memory chip. In other words, a higher level language makes considerable demands on the hardware and may require a long time to execute. Because of this reason, an assembly or machine language is still usually used for microprocessor applications. (With a machine language, every programmer instruction has only one corresponding individual machine instruction, and machine language programs optimize hardware. Obviously, writing in machine language is labour intensive and very expensive.)

Testing involves the examination of both the hardware and software efficiencies and how the two merge in a prototype. Testing occurs with "application development systems". As one study has noted:

> "Applications systems are not yet, as their name might suggest, software engineering tools. They are essentially test instruments with two functions -- to simulate the behaviour of a design system in response to real environmental signals, and to emulate the signals themselves in a circuit and check whether the prototype behaves satisfactorily in these artificially created random situations. They therefore play an essential role, especially in the development of mass market applications (which must be able to function under a very wide range of conditions over their lifetimes and would be difficult to maintain, e.g. in vehicles, measuring instruments, toys, etc.), or high risk applications (air control systems, aircraft, weapons, etc.)." (47)

5.1.2 The Programming of Gate Arrays and Standard Circuits

In this context, programming involves the product's overall design. Programming of standard cells and gate arrays involves stringent qualifications and the use of computer-aided design systems, which are still very expensive.

After a prototype circuit has been designed, optimized and tested, there are three methods of putting it into industrial production. These methods depend on the specific ways the software has been incorporated into the circuitry:

> "In the first method, when the circuit has a totally or partially custom hardware configuration (gate arrays and standard cells), it will be mass produced by a semi-conductor manufacturer. The circuit, generally integrating microprocessor and memory on one chip, will be made and tested in its finished form with a manufacturer. Entirely intended for one predefined application, it will be wholly and completely firmware.
>
> The second alternative is to use standard circuits, microprocessors and read-only memories (ROM'S). The ROM will incorporate the program and will be mass produced by a memory manufacturer, and combined with brought-in microprocessors, either by the same manufacturer or by the system constructor. This is another instance of firmware, since the ROMs incorporate dedicated, unalterable programs. However, if they are in a form of interchangeable plug-in cards then software alternatives, on other cards, can be introduced in the circuit (as with second generation video games).
>
> A third possibility is to use programmable read-only memories (PROM) to carry out the software. In this case the memory, being programmable, can be loaded by means of a development system or microprocessor. This type of system may, therefore, be reprogrammed several times in the course of its lifetime, but remains too costly and is used mainly in the 'prototyping' of microelectronics systems." (48)

It may be said, without exaggeration, that to date microelectronics has permeated merely a tiny percentage of its possible applications, and the main inhibiting factor to its further diffusion is microprocessor software.

5.1.3 Sources of Microprocessor Software

Most microprocessor software was, until the past few years, produced exclusively by the chip manufacturers themselves, who would cooperate with large customers to develop products incorporating chips (watches, computer equipment, calculators, etc.). As the technology diffused and end user firms built up their software skills, software roles of the hardware manufacturers have gradually lessened, with the exception of mass produced customized items such as automotive electronics.

Firms producing microprocessor software really took off only in the mid-1970s and concentrated on writing software for end users. Depending on their skill sophistication, microelectronics software houses may produce software for either standard cells or gate arrays. Most, however, are concentrating on standard circuits, and only about 100 software houses in the US in 1982, for example, had acquired the skills to program gate arrays. Since microcomputer software is embedded in chips, and Canada imports most of her chips, this form of software is mainly imported.

5.2 Microcomputer Software

The phenomenal growth of microcomputer software in business must first be noted. This software is being sold in a number of ways. Softsource of Dallas, Texas, for example, sells programs exclusively to owners of IBM PCs, while other stores sell software for a diversity of microcomputer manufacturers. Store locations vary from shopping centres and high density traffic areas, such as discount stores, to suburban locations placed near computer outlets. Distributors are also putting software racks in mass retail outlets and one operation had racks in more than 400 stores at the end of 1983. Other firms are also selling rack franchises. Some entrepreneurs are planning software kiosks in grocery stores and practically everywhere else. Both Safeway and B. Dalton Bookstores are also experimenting with microcomputer software sales.

The new generation of microcomputer software is considerably improved over the best software that has been written to date for mainframe computers. It is more reliable, better documented and easier to actually use. Still, business organizations are finding it extremely expensive to apply new microsoftware. Typically, a business which has perscoms might require eight to ten software packages for databases, messaging, word processing, accounting, spread sheets, etc. The actual effort spent in learning to use these packages is significant, and consequently selection of business software packages takes on a more critical dimension, not merely because of the high costs involved in acquiring them but in the time expended learning how to utilize them. Also, in spite of considerable propaganda to the contrary, little microcomputer software is easily portable from the machines of one vendor to those of another. For example, a user who wants to do a task as simple as transferring data on a single microcomputer between programs produced by two different vendors will still have significant problems. (In fact, if the source codes and input and output formats of these programs are not known, this problem will not be solvable.) There are also still major problems in transporting data between personal computers and mainframes.

5.2.1 Trends

In 1984 manufacturers of microcomputers stablized around two standards: those of IBM and Apple Computer. In a March 1984 Softcom software show in the US, more than 80% of all business and educational software was designed to run on the IBM personal computer and on IBM-compatible computers such as the Hyperion.

The IBM PC has spawned an entire industry of hundreds of hardware and software companies which grew up overnight, but a problem with manufacturing IBM PC clones is that a firm becomes extremely vulnerable to IBM's pricing strategy. For example, Columbia Data Products Inc. had 1982 sales of $9 million, which grew to $50 million by 1983, and Compaq Computer Corp., with no 1982 sales, this year will exceed $100 million. Software growth resulting from the IBM PC has been similarly phenomenal. For example, Lotus Development Corp., which manufactured their 1-2-3 program for the IBM PC had annual sales in 1982 of $30 million.

One major development here was the elimination of CP/M as the predominate operating system for microcomputers and its replacement by MS/DOS. Since software is always written for specific operating systems, with the death of CP/M as the main operating system for microcomputers, little futher software will ever be written based on it.

Another new trend apparent at the show involved relational software, i.e. software written to easily create and manipulate structured records, tables and reports.

Another is expansion software, i.e. software which modifies existing software packages. For example, Memory/Shift is a software package which divides a computer's memory into several segments, each varying in size and each which can then run diffferent programs. A user, in other words, would shift from one memory segment to another, thus from one application to another, simply by pressing a key.

Another new trend is idea-processing software, which allegedly supplants word processing. (This is actually just another varient of report generators, and one typical product, for example, Think Tank, of Living Videotex Incorporated of California, facilitates the organization and manipulation of outlines in point form before the actual written text.)

Perscom-based office automation software is headed toward open network architectures, i.e. software systems which attempt to connect computing technology of several vendors, and software which enables perscoms to link with mainframe computers.

One trend involves more comprehensive software, programs incorporating several functions, i.e. data base management, general ledger, etc. In fact, when a customer has different vendor's software for different applications, none of the software usually works together.

Another development involves multi-window software, i.e. an operating system which allows computers to run several off-the-shelf applications from different publishers at the same time. For example, using Desq of Quarterdeck Software Inc., a user may load common off-the-shelf packages and run them at the same time, each program having its own visual window on the video display terminal. In some cases, data may be moved from window to window.

Several software companies in fact are now offering products containing "windows". Visicorp, publishers of the massively successful Visi series of business programs, began shipping Vision, a windowing package, in December 1983. Also, Microsoft, the leading 1983 perscom software publisher, introduced a program called Windows.

These are not new notions. The idea of an electronic mouse was introduced into Xerox's workstation, Star in 1981, and Apple further refined this idea in its Lisa and MacIntosh computers. Similarly, in October 1983 IBM introduced a perscom which had windowing capabilities. However, now a number of software firms such as Quarterdeck are writing windowing programs for many computer manufacturers to lend windowing capabilities.

The two main competitors in windowing software, Visicorp and Microsoft, have followed very different directions. The first firm spent more than $12 million over a three year period to develop their window display system, retailing for $495 US, and also makes programs which run in this system. Microsoft has built just the framework for windowing rather than the actual programs which run in it.

Other firms, including Lotus Development Corp., are all revising versions of their software programs to fit into windows.

5.2.2 Window Standards

The three main producers of software windows -- Visicorp, Digital Research and Microsoft -- are all attempting to make their product an industry standard. It is expected that within a year windowing programs will be a standard feature of all perscoms, but the personal computer marketplace is not big enough to support several different versions of windowing software. Also, many of the applications packages such as word processing will necessarily have to be rewritten to work within each windowing package. Consequently,

> "The battle is to establish whose windowing software is going to win, because software developers don't want to write programs for eighteen different systems ... the pressure to standardize will be there." (49)

Microsoft may have the edge because its MS/DOS operating system is widely used as a standard on perscoms, and its "Microsoft Windows" is a modification and extension of MS/DOS. Visicorp has taken a different tact and is emphasizing applications programs that can actual run in windows, in addition to making the windowing program. Digital is also making applications software which will run in conjunction with its new windowing product.

5.2.3 Personal Computer Software Purchases

In 1983 the average perscom owner in the US bought $608 worth of software at the time of perscom purchase, and during the first year of ownership spent an additional $439 on additional software.[50] Given such additional software spending, Frost and Sullivan of New York predict a $1.2 billion market of micros software by 1985. This secondary purchasing of software is important, since it has been theorized that after the first excitement of buying a perscom, many users do not purchase much additional software but often copy the rest. Frost and Sullivan's buyer survey disproves this conjecture.

In this survey, one-third of all the perscom software was purchased through the mail, and approximately one-third of the software was purchased from computer chain stores, with independent specialty retailers accounting for 22%. Only 1% of the share of software sales occurred at book stores.

Owning a diskette containing computer software without having good, printed documentation makes the software totally useless, and a key factor noted in the Frost and Sullivan study pertaining to future perscom software marketing will involve pre-offering documentation for sale before offering the actual software, so a tentative buyer can examine the product in advance.

5.2.4 Perscom Standards: Unix and MS/DOS

On January 12, 1984, IBM announced that purchasers of their PC could use two operating systems, the presently-available MS/DOS, and for an extra $900 they could have Unix.

Unix is important because it does not depend on the specific layout of the computer's circuits. This requires a brief explanation. Any computer operating system is essentially a mediator:

> "... between what the computer user wants the machine to do (the particular application such as word processing that he has in mind) and the circuits that actually carry out the computations. The circuits, although very fast, can function only at a simple level of being either on or off ... an elaborate series of translations has to be performed to get from the English (or numbers or pictures) on the user's computer screen down to ones and zeros. Among the things it does, such as keeping track of where the information is filed on memory discs, an operating system performs those translations." (51)

Most operating systems are awful, haphazard quilts, and although they accomplish the job, each one depends on the specific design of hardware circuits they were made for.

Unix, however, developed by AT&T's Bell Labs in 1970, has a simpler and more logical structure, is more independent of circuitry, and it's inventors also created a "higher level computer language" specifically designed for Unix, called C. Unix was designed to help software engineers write specific application programs quicker and easier.

Since it is written in a higher level language which itself has a layered structure (C), when Unix is moved from a perscom of one manufacturer to that of another (based upon entirely different microprocessor chips), only a small part of the Unix program has to be rewritten, and the end user can actually change computers and have quite a lengthy application program run on them.

Secondly, Unix can handle several tasks simultaneously by multiple computer users. This is important when hooking perscoms into large mainframe computers.

5.2.5 Perscom-Mainframe Linkage

Most early software was designed to accomplish specific tasks only on the computer the user was working on. However, a number of new packets, such as Informatics General Corp.'s Answer/DB, allow mainframes to shoot data to perscoms in formats available for immediate use, such as spread sheets. This is difficult because a perscom and a mainframe will usually utilize different means of storing and formatting data, and differing commands for retrieving it. It would thus seem likely that much mainframe and micro software in the immediate future will necessarily have to facilitate such linkages.

Exhibit 5-2 presents the main microcomputer operating systems, the producing firms and 1982 sales. Of special interest is the Pick operating system which has adapted software designed initially for mainframes and minis. Unlike most developers of system software, who make it available to computer manufacturers for a small up-front fee plus a royalty on each machine sold, Pick does not charge a royalty but a one-time licensing fee of $1.1 million.[52]

5.2.6 Perscom Software Marketing

Many of the major problems for Canada's multitude of small software houses involve the high cost in competent and clear documentation, software packaging and marketing. Most small Canadian firms don't have resources to adequately promote and market their products. International software marketing is expensive and risky, and several small Canadian firms are specializing in marketing of other firms' software rather than develop their own. For example, Data Kinetics seeks international sales and provides technical improvement to small firms with one or two software products. With a revenue of around $4 million annually, this firm employs more than 50 people, with 300 to 400 worldwide clients.

In Canada several industrial associations concentrate on awareness of software marketing and export needs and clarify legal issues surrounding software production and sales. Most firms have scarce funds available for market research, and persons with technical backgrounds are often notoriously bad at marketing their products. Software writers are no exception.

EXHIBIT 5-2

Microcomputer Operating Systems

Name	Company	1982 Sales
MS/DOS	Microsoft	$32 million
CP/M	Digital Research	$15 million
Unix	AT&T	-
UCSD P-System	Softech Microsystems Inc	-
Pick Operating System	Pick	$5 million
Oasis	Phase One System	-

Many of the Canadian software houses have formal or informal subsidiary relationships with American hardware manufacturers and often manufacture software for a specific line of hardware, e.g. Hewitt Packard minicomputers. These companies sometimes try to sell an entire system composed of the parent's computer plus the subsidiary's software and attempt to gain revenues by customizing and maintenance of the software. However, packaged off-the-shelf software can be just as good and a whole lot less expensive.

Some US hardware manufacturers and their subsidiaries, e.g. Prime Computer of Canada Ltd., a US subsidiary, will attempt to get Canadian firms to write software for their computer. Prime has given free machine time to Canadian software developers in Calgary (for work in reservoir modelling) and to Accugraph Corp. of Toronto (for CAD/CAM software).

A number of analysts have claimed that the small entrepreneurial phase of the perscom software industry has ended and that we have already moved into a phase in which only professionally managed and marketed companies will be successful.

However, there has been a growth also of independent micro software marketers. Simon and Schuster Inc., CBS Inc. and B. Dalton Bookseller Inc., in addition to American Express, are all planning major programs for software distribution and marketing, including longer term planning to use optical storage cards instead of disks. American Express will distribute well known programs through the mail and will offer software support via toll free 800 numbers. The book publishers, on the other hand, have contracted with individual programmers to develop software which will be distributed through existing publication channels. Most of these newcomers to the software industry are not producing their own software but are publishing software written by others, concentrating almost exclusively on distribution/marketing problems. The president of Software Publishing Corp., has asserted that:

> "The problem of who survives the software business simplifies to who controls the distribution." (53)

No software company can now produce all of the applications demanded by a world market, and one trend we saw is for large software firms to purchase smaller specialized ones. Such mergers are also beneficial for the smaller firms because marketing ability is increased.

If control of distribution channels becomes a central factor in software markets, it is quite possible that major North American book publishers will have a real edge. Given the present structure of the Canadian publishing industry, this does not bode well. But to date publishers still have an insignificant piece of the market, and a number of skeptics have pointed out that there are major differences between book and software publishing, that software, for example, must be continuously updated for new computer technologies entering the market. Also, system software must be virtually error free.

5.2.7 Rentals

One new US trend involves stores for perscom software rentals, from which consumers can rent and copy any perscom packet. Software rental companies typically charge fees for a week's rental of the packet of about 5%-15% of its retail price. Thus a customer might rent a $550 packet for $20 or so, make a copy of it and return it. Rental chains have also sprung up in Tokyo. With estimated annual losses of $500 million, US software producers are looking for protection from the rental stores, and in May 1982 Micropro International Corp. sued United Computer Corp. for piracy. (Micropro developed the word processing packet WordStar, which presently has almost half of the world perscom word processing software market.)

The main legal issue involved in the suit involves the present licensing system of the US software industry. This was developed by IBM and has been adopted by most software producers. Accordingly, when one buys a software packet, one is purchasing merely the right to utilize this packet, with ownership remaining with the producer. Accordingly, any user cannot legally sell, reproduce or give a copy away free. Micropro's position is that since United is not an authorized Micropro dealer, it is therefore an end user and does not possess the right to sell or lend to other people.

Another issue involves software copyright. It is difficult for Microsoft to claim that rental of their packets involves copyright violations, since the US court has already agreed that the real aim of United's rental is copying.

In related developments, last year Apple sued Franklin Computer Corp., alleging that the latter had copied Apple's operating system. A federal judge, however, denied Apple's request for an injunction which ordered Franklin to stop making their imitations and ruled that operating systems are not necessarily designed to "explain or communicate information" which is the US test of whether or not a work may be copyrighted. (More recent copyright issues are subsequently discussed in the policy sections.)

5.2.8 Micro Software Searches

By conservative estimates there are now more than 35,000 software packages available and a wide variety of perscoms now on the market. Often vendors and users are not fully aware what is available. Consequently another type of firm called the software searcher has emerged For example, Travellers Insurance Co. in 1983 created a special division called The Information Centre to locate software for their planned 3,000 perscoms which will be distributed to field agents.[54] Typically, employees might borrow programs from such a centre and then test them and report back whether or not they should be improved for general corporate use. The benefits of centralized evaluation of software in large firms are obvious.

To aid users who don't work in large firms, a number of entrepreneurs have started software market search services and software directories. Given the rapidity with which new applications programs are appearing, a number of firms have also started monthly magazines, such as List, which contain software critiques.

However, many software packages have a product life of six to eight months, and often evaluative information is a year out of date by the time it gets published.

To make more contemporary useful evaluations, firms such as Softsearch use computerized files to evaluate and present information on 30,000 software packages. This service already has 10,000 clients in the US, who pay $175 to subscribe to Softsearch's library. Most of the users of this service are individual perscom owners or businesses, with the remainder being computer consultants or retailers. Such services are more useful for inquiries into unusual software, but for most applications such as word processing, firms are finding that directories are more cost efficient.

A number of these software search firms also provide consulting services on using applications packages. For example, ITM, a software search service in California, charges $100 per year, and users can call ITM's consultants for personal guidance. A number of retailers are also installing terminals in stores so that customers, with in-store supervision, can actually try software before they purchase it. For example, PC Telemart Inc. in Fairfax, Virginia, is installing computer kiosks in retail stores and charging the stores $500 per month. A customer calls the store the day before he "test drives" the software, which is downloaded from PC Telemart's central computer, and there are plans to go on-line if this project is successful.

PC Telemart also opened a National Software Library in mid-1983, where computer and software retailers and others can test drive programs.

5.2.9 Software Licensing

A lucrative way to gain software profits is through licensing. For example, the main operating system for perscoms, MS/DOS (microsoft disc operating system), invented by Microsoft of Bellevue, Washington, is now the new perscom standard and has been licenced to most manufacturers in North America.

Any hardware manufacturer competing with IBM now wishes to advertise "IBM compatability" and to achieve this compatability, they have to licence MS/DOS from Microsoft. It has been estimated that to date the company's licensing revenues from this program are in the range of $10 million. Microsoft also was awarded the contract in to supply the operating system for the poorly selling IBM

PC Junior.[54] Microsoft, in fact, now provides this operating system for more than 90% of all IBM computers and IBM compatible computers.

Recently Microsoft has switched to applications software. (A computer manufacturer pays approximately $10 per computer for a typical operating system, while new applications programs sell to retailers for between $40 and $250.)

5.2.10 The Canadian Microsoftware Market

Predicting software markets, by computer type, is a tenuous activity at best. Data from several large, reputable marketing firms may vary significantly, as the astute reader will have noted, given different methods of surveying and differing product divisions. However, it is clear that the market for mainframes is a mature one and that the microsoftware market is growing. Exhibit 5-3 presents estimates for Canada and the US of the microsoftware market.

5.3 Industrial Automation Software

Automation software governs, and runs on, robots, numerically controlled machine tools, production and testing equipment, and computer-aided design and process control equipment.

Increasingly in the 1980s microprocessors are serving as dumb brains for machine tools and industrial process machinery. Through simply altering programming instructions, a machine tool may be changed to another set of instructions and thus increased production speed. Although a rapid diffusion is expected here during the next five years, by 1982 the growth of computer controlled machine tools in the developed economies was slight, with the existing stock of such machine tools being around 5% of the total number of all machine tools.[55] In Japan, in contrast, this number is growing at rates of between 25% and 35% per year. One study done in Canada[56] estimated that by 1985 there will be 3,200 new machine tools installed, most of these being computerized. The majority of vendors, this author notes, have discontinued the manufacture of hard-wired numerical control systems for machine tools and now produce only computerized drafting systems which are

EXHIBIT 5-3

Canadian Software Revenues by Type of Computer
($ millions)

	1981		1986	
	$	% of Total	$	% of Total
Micro	32	5.3	217	9.3
Mini	131	21.5	594	25.3
Mainframe	445	73.2	1,535	65.4
Total	608	100.0	2,346	100.0

Source: Evans Research Corp., as cited in Touche Ross & Partners, "Study of the BC Software Industry," Vancouver, March 1984, p.13.

microprocessor based. Also, a number of large Canadian aerospace manufacturers have been considering "distributed control" in which a single large computer is given design instructions by engineers at terminals and then feeds these into a series of microprocessor-based machine tools.

Just as the productivity implications of computers and communications became apparent only when they blended, the same is true of the blending of robotics, CAD/CAM and microprocessor controlled machine tools.

Exhibit 5-4 presents growth estimates for the US market for industrial electronic equipment from 1981 to 1986. New, more flexible automation systems will soon involve the automating of batch processing, and the main bottleneck now in industrial automation involves software which will adequately exploit new programmable equipment. Robotics technology, relying heavily on AI-related visual, aural and tactile sensing, is in an ill-developed primitive state, and a main inhibition for the further diffusion of CAD/CAM is the lack of software which enables engineers of CAD modules to direct CAM equipment. (In 1981 there were only 450 industrial robots in Canada.)

A 1982 US survey[57] showed that most numerically controlled machine tools being used by US machine tool manufacturers necessitated, on average, 70 hours of programming monthly per machine, and if we assume that the lifetime of the machine is five years, this involves 4,000 hours of programming (which would soon cost much more than the original hardware).

In the area of robotics, although scant information is available concerning proportions of software to hardware costs, it is frequently estimated that software comprised 50% of total development costs, and this estimate includes only the system software written by the robot designers and does not include applications software. With the next (third) generation robots (semi-intelligent robots which can partially program their own applications utilizing sophisticated AI techniques), it is expected that the software development costs will be over 90% of total costs.

EXHIBIT 5-4

Industrial Electronic Equipment Markets in the United States, 1981-1986 ($ Million)

	1981	1982	1983	1986
Numerical control systems	198.5	265	371.7	998
Automated testing equipment	969.3	1,108.6	1,421.1	2,676
Measurement systems	197.7	239.6	269.1	394
Process control equipment	1,288.6	1,481	1,604.1	2,529
Computerized energy management equipment	440.5	571.2	721.3	1,550
Industrial robots	191.2	217.1	290.2	537
Computer-aided design systems	312.4	450.9	744.6	1,801

Source: Electronics

5.3.1 Computer-Aided Design

The main cost, more than 50%, of developing computer-aided design systems involves software and similarly for process control technology, i.e. the automation of continuous production processes in industries such as petrochemicals and steel. These industries have undergone significant productivity advances with the introduction of robotics technologies such as sensor/microprocessor combinations. These have necessitated an increased requirement for real time software which is usually built directly into the equipment and dedicated exclusively to a single purpose.

New software will also be required at various levels in industrial automation: first to link up CAD/CAM systems. The actual integration of CAD/CAM has only occurred in a few instances, e.g. in numerically controlled machine tools and the automatic generation of production masks. With the exception of chip production itself, the linking of CAD/CAM is still in a very primitive stage in other industries, but there is considerable research in this area, including integrated computer-aided manufacturing projects of the US Airforce, and research at the Universities of Hokkaido, Tokyo and Kobe.

Naturally enough there are a number of attempts to standardize the linkages between CAD/CAM. For example, in the US, the Electronics Industry Association is trying to make software for CAD systems portable, irregardless of the numerical control machinery.[58]

Another task of software in industrial automation is to hook automated machinery, such as numerically controlled machine tools, together. The technology for such a "flexible manufacturing system" presents horrible software problems as the software must solve compatability problems which always arise from hooking together control systems of different manufacturers and must manage highly varied and complex combinations of individual pieces of machinery, which themselves are driven by complex software, and must be able to rapidly call forth differing parts and tools. Although most of this software is manufactured in-house, some software for flexible manufacturing is already on the market, e.g. GEF produced by General

Electric, Data Highway by Allan and Bradley, PC Link/1000 AD by Hewlett-Packard, and Tipros of Toyota. Almost all of Canada's industrial automation software is imported, usually in the form of firmware.

5.3.2 Industrial Trends

In spite of the above problems, a central industrial trend of this decade is the development of CAD networks. In this case the software problems involve all of those discussed above plus networking the software.

Another trend of automation involves integration of computerized management functions with those of computerized production. In 1984 two North American firms, Honeywell and Xerox, announced software which reportedly integrated production management and control. Honeywell's TDC system will allow interaction between plant level automation and management functions on mini computers previously functioning independently, and the Xerox Manufacturing System also integrates some managerial and production control functions. Other new systems involve the interactive manufacturing control system by NCR, the manufacturing system of Burroughs and IBM's Copix. In this context, local area networks which have hitherto been prevalent mainly in business are spreading to factories, and local communication linkages in factories are becoming extremely important. It is thought that factory local area networks will be able to integrate data for a large number of functions. The Yankee Group has predicted that the market for local area networks in factories in the US, which in 1982 was only $35 million, will have grown to $65 million by 1985, and although manufacturers continue to differentiate their products by different standards, the IEEE in the US has been working on problems of standardization of local area networks in factories.

5.3.3 Production of Automation Applications

Applications of automation software have a very small lifetime (equivalent to the lifetime of the product) and obviously most may not be economically packaged. However cost considerations have forced some productivity improvements in the production of automation applications software. A number of languages, for example, have been developed specifically for numerical control. One of the first of these langauges, automatically programmed tools (APT), was developed in 1959 at MIT and is still in use today. Other languages involve Compact II of Manufacturing Data Systems Inc., Promo invented in France and Fanuc's FAPT.

There is even a wider diversity of robot languages, given the increasing diversity of robot producers from differing countries, and given also the diversity of control systems used in robotics (continuous, point-by-point, etc.). Although until now most robotic systems have been programmed via simple learning (the replication of remote control systems), with the diffusion of second and third generation robots, programming by means of higher level languages will become the normal case.

A number of robotics researchers and manufacturers have developed robotics control languages -- such as Stanford University's Acroymm, VAL of Unimation and IBM's AML -- but none of these languages are in general industrial use. In Europe a control language standard is growing out of research done at the Institute of Applied Mathematics at Grenoble. Researchers there have developed a language which can guide four types of robotic movement -- linear and diagonal movement, movements which are subject to sudden halting and movement which can be guided by sensory data. Itmi and other firms are commercially producing robotic software based on this language, which has been adopted by manufacturers in Italy, England and France.

5.3.4 Production of Automation Systems Software

Original automation equipment manufacturers are conducting extensive research into the architecture, purposes and communication interfaces of systems software for automation. Most of this work involved in design and specification is still written in Fortran. This software also is often embedded into hardware, sold in very large North American markets, and there is as yet comparatively little competition (say compared to systems software for data processing).

5.3.5 AI and Industrial Automation

It is generally expected that the most significant advance in industrial automation over the next decade will involve the increased use of AI techniques to design applications software. Increasingly, AI techniques will be used in industrial production to interpret visual and tactile information and program instructions. A number of computer algorothms are being used, for example, to recognize patterns. However, these sometimes do not work very well on the boundaries, and increasingly AI techniques involve crude forms of "reason" in determining, for example, pattern shapes; variants of expert systems may function as system software for pattern recognition systems and may be reprogrammed for specific patterns.

5.3.6 Sources of Automation Software

Industrial automation hardware and software, we saw, is designed almost exclusively by the user industries. User firms normally buy hardware components from specialized suppliers and then put them together and program them themselves. The demand for automation software, then, arises in industries which are adopting automation equipment, and is concentrated in industries such as automobiles, electronics, etc. With the exception of Japan, however, automation hardware and software is diffusing very slowly in most countries. Although a number of inhibitory factors are involved, the main factors are equipment incompatabilities and the extent of investment in existing equipment. Given the fact that most industrial firms have a large stock of process equipment which has been designed independently and later put together, and different equipment may have different amortization cycles:

> "It is very difficult to integrate software into such hetrogeneous combinations, not designed with a view to integration. So unless the user firm is prepared for massive write-offs, it has to wait until the earlier investments have been properly amortized before it can redefine its production systems more closely in accordance with the state of the art." (60)

Although most automation software is produced in-house, some is produced by outside firms. Systems software for automation technology is even more expensive than non-automation systems software, and it's produced almost exclusively by outside firms, while automation applications, always being incorporated in the specific manufacturing processes and industrial products, is produced almost exclusively in-house by user firms.

5.3.7 In-House Production

In-house production of automation software has occurred in a haphazard decentralized way, with no main facilities for its production. Most of the in-house automation software to date has comprised simple programs for numerical control machine tools, test equipment and automation technology which may be programmed. Increasingly, however, very large firms in several nations, such as General Electric, Westinghouse, Hitachi, Renault and others, have started centralized automation centres which mainly concentrate on software for the new integrated CAD/CAM systems. Although to date these systems have mainly done in-house work, they may gradually evolve into an outside consulting automation engineering industry. There are already trends in this direction with Westinghouse and GE's centres.

5.3.8 Outside Sources

On average 50%-80% of the costs of a CAD system comprises software, and increasingly the manufacturers of automation hardware are embedding larger amounts of systems software in computer-aided design modules, robots and machine tools, etc. At the same time, the computer manufacturers themselves have diversified into software for computer-aided design and production management. Increasingly, systems software, with the hardware merely as a vehicle, is the main selling point and involves the main development costs for diverse automation equipment ranging from new languages for robots, to system software for numerical control systems, etc.

Even so, the hardware manufacturers have not yet unbundled their systems automation software, as other software has been unbundled. However, as we have seen, large manufacturers of specialized automation equipment have started separate engineering services, and much of this work involves the writing of automation applications software for end users. This is almost the universal case with robotics, and as this industry is increasingly unbundled, a new software industry will be developing.

Given the advantages of the large manufacturers with integrated software/hardware automation systems, the automation standards issue becomes important, since large manufacturers are attempting to make their technology de facto industry standards (e.g. Computervision). On the other hand, smaller manufacturers of automation equipment are attempting a more organized standardization effort.

5.3.9 Specialists in Automation Software

A number of firms have specialized solely in applications of automation software for end users. Providing design and engineering consultancy services in addition to software, this work still involves custom software for specific applications. Clients are mainly small manufacturers.

Other firms have specialized in automation systems software and usually specialize in one or two products. These firms are either original equipment manufacturers and incorporate software into hardware which they have purchased or offer software directly to end users. Examples include a Schlumberger subsidiary, Manufacturing Data Systems Inc., Dynax in Japan and Structural Dynamics Research Corp, a subsidiary of General Electric.

Finally, in several countries, industries and universities with research laboratories have proved the prime source of technological innovations in automation software, and such research laboratories play a much more important role in software than they have in hardware development.

5.3.10 The CAD/CAM Market in Canada

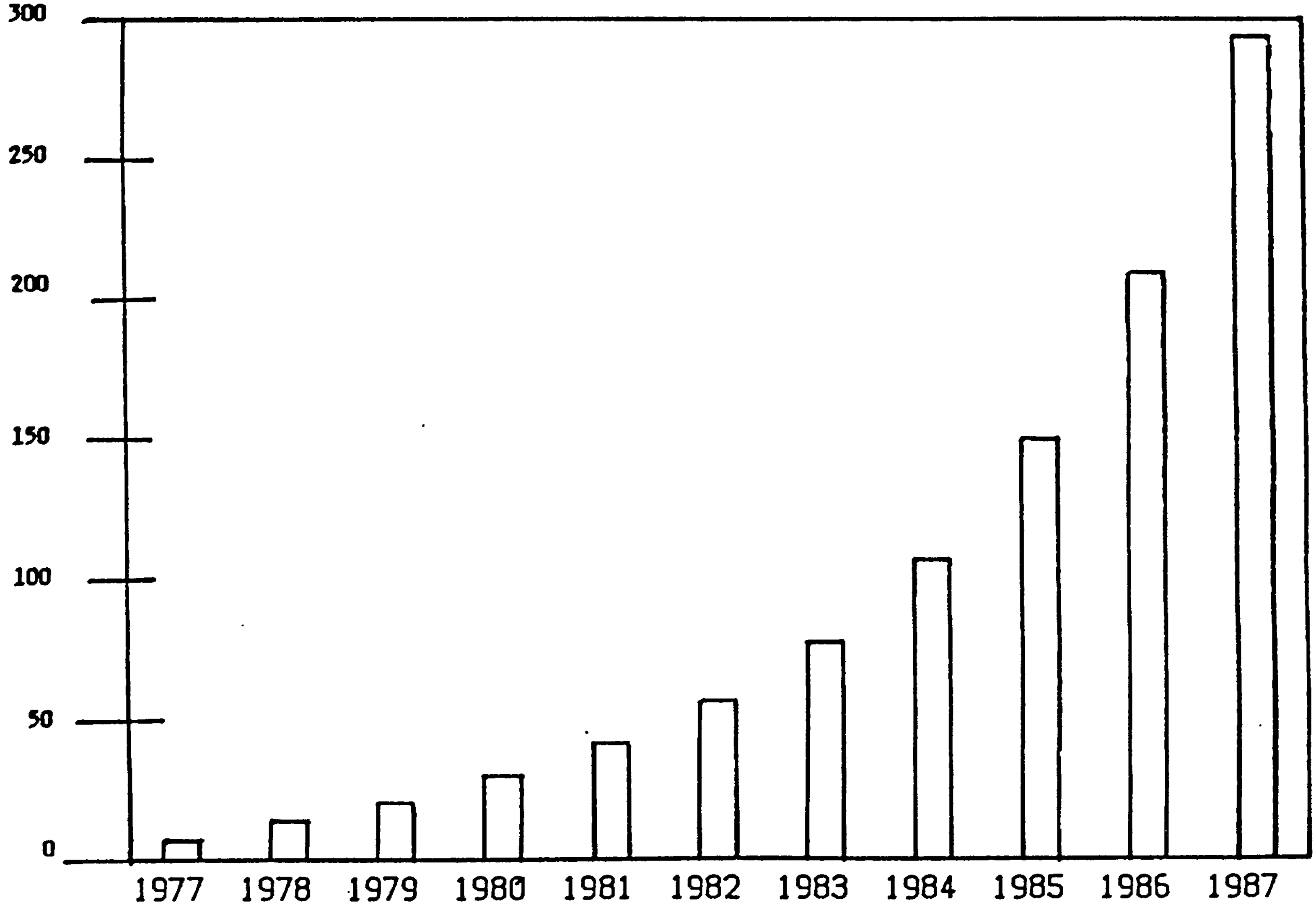
Several analysts anticipate the Canadian CAD/CAM market, which was $57 million in 1982, will grow by 30%-35% annually. Exhibit 5-5 presents estimates for Canadian CAD/CAM growth. The Canadian market is dominated by US turnkey vendors (Exhibit 5-6). This group controls approximately three-quarters of the Canadian domestic market share, and this domination is expected to continue since turnkey vendors offer a rapid cost-efficient way to implement this technology. The main suppliers are Applicon, McAuto, Computervision, Auto-Trol and Calma. Exhibit 5-7 presents estimated growth rates for specific application sectors until 1987.

A main market trend is the introduction of 32 bit machines into the turnkey market. Although these systems right now have comparatively little application software and can support relatively few users, both of these factors may change in the near future.

Other reports, such as a private 1983 study by SRI International, have estimated the 1988 total North American CAD/CAM market will be around $4.4 billion, with approximately 36% of this amount being captured by (non-embedded) software.

EXHIBIT 5-5

The Canadian CAD/CAM Market To 1987



Source: Evans Research Corporation Surveys & Estimates

EXHIBIT 5-6

Market Share of Key CAD/CAM Competitors

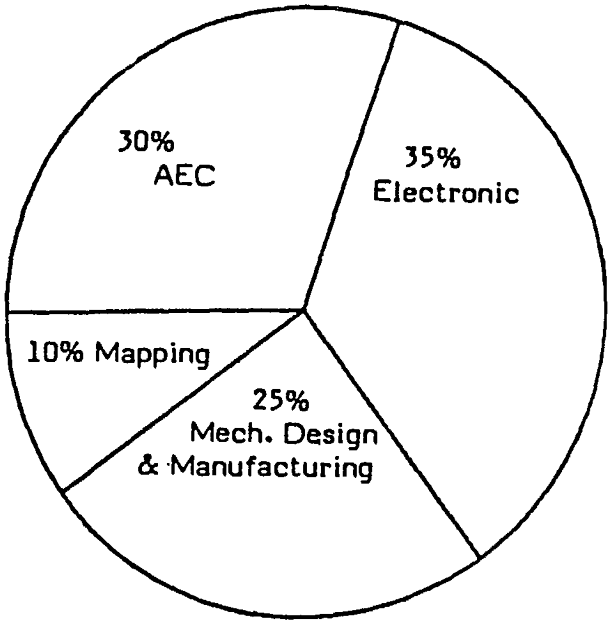
	1982		1983	
	Revenues ($)	Market Share (%)	Revenues ($)	Market Share (%)
Turnkey Vendors and Others:				
Intergraph	$ 14.0	26	$ 18.0	23
Computervision	6.9	12	8.6	11
Calma	5.0	9	7.0	9
Auto-trol	2.5	4	5.0	5
Applicon	1.0	2	3.0	4
Omnitech	1.8	3	2.8	4
Orcatech	1.3	2	3.2	4
Calcomp	2.1	4	2.8	4
Systemhouse	1.8	3	2.4	3
McAuto	1.4	2	2.1	3
Others	3.9	7	3.6	4
Total	$ 42.7		$ 58.5	
Hardware Manufacturers:				
IBM	$ 6.1	11	$ 8.2	11
Digital Equipment	2.1	4	2.6	3
Prime	1.1	2	2.3	3
Hewlett-Packard	1.7	3	2.2	3
Data General	1.3	2	2.1	3
Others	2.0	4	2.1	2
Total	$ 14.3		$ 19.5	
TOTAL MARKET	$ 57.0	100	78.0	100

Source: ERC Surveys and Estimates

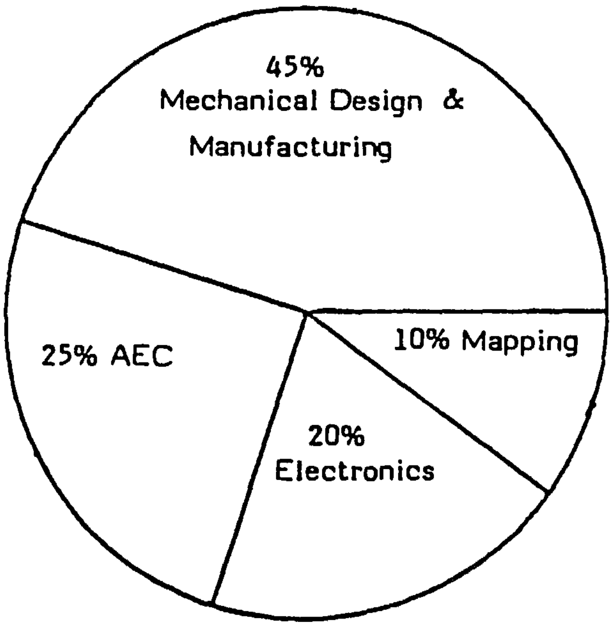
EXHIBIT 5-7

Market Share for Major Application Areas

1982



1987



5.4 Educational and Training Software

Early, simple applications of computer learning (CL) presented material almost mechanically, in a standard fixed sequence, using the computer as a textbook with an electronic screen. Instead of fixed sequences, CL can now incorporate "conditional branching". A number of alternative sequences are included in a program and depending on the individual's response, different sequences are entered. This allows for "individualization" in the learning process. More advanced systems build up "learner profiles" through testing and monitoring that can take individual learning patterns into account.

More sophisticated systems incorporate artificial intelligence approaches. First, expert systems are used in which the computer is programmed with knowledge in a particular domain. (The SOPHIE system tutors in electronic troubleshooting, while SCHOLAR is an expert on world geography.) Secondly, intelligent systems expand the "learner profile" idea into a more complex "student model" representing the hypothesized knowledge state of the student. Based on this student model, the computer selects the best procedures to present material. The BUGGY system, for example, can diagnose student problems (or "bugs") in doing simple arithmetic.

The trends in CL applications that stand out at this time are:

1. Visual environments are increasingly interesting, with graphics, animation and the use of full video with videodiscs.

2. Applications are becoming more involved for the learning, using simulation, animation, "socratic dialogues" with expert systems and so on.

3. The use of intelligent systems is advancing steadily. (Even in the microcomputer market, where computer capacity is limited, AI is making inroads. Microsoft, for example, is planning to market in 1985 an AI-based tutorial to teach software writing.)

5.4.1 Components of CL Hardware

Putting the computer to educational or training use requires hardware, software authoring tools to create content, and content or learning applications.

The hardware used to provide computer power in CL at present is mainly minicomputers and microcomputers. As is the case with computing generally, the trend in hardware has been towards shrinking size that holds increasing capacity. Mainframe examples in computer learning still exist: the PLATO system, an early CL giant, continues to use a timeshared mainframe computer model; however, prospective customers faced a price of some $2M for a CYBER mainframe or substantial ongoing changes to connect to PLATO centres. At the other end of the size spectrum, developmental work is inching towards microcomputers the size of a book.

Minicomputers today fit most requirements for substantial computing power where, for instance, a CL language demands large computer resources, or the system has to support a large number of users. Prices vary but always involve a considerable expenditure.

The low cost of microcomputers has caused a rush of buying in elementary and secondary schools, and micros are also spreading steadily in the home and small business market.

Microcomputers being purchased today can be embellished with a number of features. Most are now highlighting colour graphics, and a variety of input devices besides keyboards can be found ("joysticks" which one rotates to move objects easily on the screen, tablets sensitive to touch or to special pens, and so on). Video interfaces can control disc players; speech synthesis can be used for spoken output and speech input capabilities are slowly developing. Sound synthesis can be used for learning or composing music; and real time clocks can set timers for gaming and simulations.[62]

Handheld units such as "Speak and Spell" are also being used, and major US and

Asian developmental projects are targeting handheld educational units at present. Since the early seventies, the Smalltalk project at Xerox PARC[63] has aimed to produce a handheld computer with high resolution graphics. Basic research on Smalltalk has been influential in a number of advances in computer technology to date.

5.4.2 CL Software: Authoring Tools

There are two types of software in computer learning: applications software (in this case the actual learning content programs for the individual), and system software, including programs to make the authoring of content easier.

CL content creation can require between 200 and 300 hours of development to produce one hour of an actual CL application.[64] "Authoring tools" for quicker creation of CL content are of two types: author languages and authoring systems. Authoring tools make content creation faster and easier, and a number of author languages and authoring systems have been developed.

An author language is a high level language for the specific purpose of making CL content. It renders the programming required for a CL application an easier task. An authoring system is a high level software interface which similarly speeds up production of CL content but does so by minimizing the programming required. (For example, an author language would have a range of commands for the detailed specification of lesson components, while with an authoring system, such specifications would not be required—the system itself would supply default conditions or prompt the author for information in small steps.) A goal of authoring systems has been to allow authors to make CL applications with little or no programming expertise.

The list of author languages includes TUTOR (for the PLATO system), the National Research Council's NATAL,[65] which has been implemented on several computers, and PILOT, available on a number of microcomputers. Examples of authoring systems include CAN's EASSy[66] programs and TICCIT's APT.[67]

Authoring systems were developed from author languages to aid in content creation--handling input of text, graphics, audio and other material. Later, authoring systems incorporated lesson definition, with simple ways to specify lesson structure. Authoring systems minimize programming in several ways: through high level commands (which could be equivalent to multiple commands in an author language); through prompting of the author; and through the use of a predetermined form of instruction. TICCIT, for example, has an instructional logic built into the system. Within this framework, the author simply provides content. The Authoring Procedure for TICCIT (APT) has enabled non-programmers to produce an hour of content with 100 hours of development, half the time a programmer would otherwise require.[68]

Some authoring systems have achieved fairly widespread use, such as the PASS system, developed by Bell and Howell. PASS uses a predefined authoring format, prompts the author in ordinary English and will cue the novice through the entire authoring process. It also contains automatic branching and grading and has capabilities for colour graphics and videodisc and tape. Most importantly, PASS runs on an Apple microcomputer and requires no programming experience.

The limitations in current authoring systems, however, are many. Most basically, authoring systems typically use prespecified patterns or "templates" for the CL material. Many claim that this works against variety and creativity. Author languages, while they require more programming, nonetheless offer greater flexibility. The tradeoff lies between flexibility and ease of use.

Secondly, authoring systems are weak in handling a number of colourful and useful enhancements to text-based CL. Simple methods for creating graphics are often lacking, as are convenient means to handle multimedia materials such as the videodisc. The use of simulation at present is limited to authors who can program; authoring tools could help define input conditions, outputs, and causal links in the simulation program.

Thirdly, there are few authoring systems that handle knowledge-based or artificial intelligence approaches to CL, in which learning occurs as a dialogue between the student and the computer. In these systems, content and instructional strategies take the form of semantically related concepts and inference rules. Such systems require complex associative databases which are laborious to make, so that authoring tools could be particularly productive. Such tools could encourage the use of knowledge-based CL.

Contradictory views on authoring tools partially stem from the fact that needs for authoring can vary widely. The question is: who is authoring and where? The elementary school teacher who wants to write lessonware needs a simple, low-cost system that runs on a perscom and may not want to learn programming. In contrast, in industry, in-house programming expertise may be available, and the value of an authoring system would lie in the extent to which it speeds up production without limiting programming options.

The following sections examine two Canadian CL languages and an influential project in computer software generally.

5.4.3 NATAL (National Authoring Language)

In the early seventies, many languages and computers were being used for CL, and content sharing was impossible. NATAL's goal was a standard language which could be implemented on a variety of computers to permit the transfer of content from one place to another.

NATAL was designed as a high level language to increase a programmer's productivity and to create transportable content. Fewer lines of code are required to produce a CL application in a NATAL program than is the usual case. Using NATAL a writer produces a detailed course design, and the burden of executing this design is transferred as much as possible to the computer.

A NATAL prototype, first implemented at the NRC on DEC computers, was transferred in 1979 to Honeywell, which was charged with a number of tasks to commercialize NATAL including: upgrading the DEC prototype to commercial standards, implementing NATAL on IBM hardware and on Honeywell Level 6 minicomputers.[69]

Honeywell Ltd. is a subsidiary of the US firm, Honeywell Ltd., which also owns part of CII Honeywell Bull. While a Canadian hardware manufacturer might have been preferred, no Canadian firm then made the size of computer that NATAL requires. From the NRC's view, Honeywell offered the possibility that its Canadian company could become a centre for computer learning in the large Honeywell complex, acquiring world product mandates within the Honeywell group. Honeywell also had the necessary money to sustain NATAL's growth pains and to launch major marketing campaigns—as it has done with CAN, discussed below.

To date, however, Honeywell has not produced a commercial release of NATAL; and in the meantime, MicroNATAL, an implementation of NATAL that runs on IBM personal computers and other microcomputers, has been developed by a Canadian firm specializing in CL, Softwords. PLATO and TICCIT have emphasized microcomputer capabilities in recent years as well.

5.4.4 CAN (Completely Arbitrary Name)

Work on the CAN language began in 1967 at the Ontario Institute for Studies in Education, with the original aim of developing an easy-to-use authoring language. The CAN language is now used for general programming purposes, and the CAN operating system supports other software functions in addition to content authoring. The main CAN software includes: the CAN-8 Authoring Language; EASSy (Educator Automated Authoring Systems), a content creation facility that generates either CAN-8 programs, Telidon databases, or a program master for optical videodiscs; the CANNET network processor, a communications system that links CAN-8 computers; and GAP, a Graphics Authoring System.

Because CAN has been used for actual content development projects in Ontario over the years, considerable CAN content is available. The Individualization

Project, for example, produced a number of college level courses, and a project in Ontario secondary schools created CL material for over 30% of intermediate mathematics curricula. This content can now be acquired on a commercial basis from OISE^ The ability to network by telecommunications between different campuses has also been emphasized.

The rights to market CAN in Canada and the US have also been acquired by Honeywell Ltd., and CII Honeywell Bull will market CAN in Europe. CAN-8 and its software options are implemented on the smaller of the Honeywell Level 6 minicomputers, and a system to support 128 terminals would cost about $100,000.

5.4.5 Smalltalk

In 1971, a long-term research project was begun at the Xerox Palo Alto Research Center (PARC), to produce the Dynabook, a flat notebook-sized package with high-resolution graphics and a touch sensitive flat screen. Design work at Xerox PARC has been seminal in the computer industry.

Smalltalk is the programming system designed as part of the Dynabook effort. It is a unique language in many ways, not least for its underlying philosophy, which emphasizes flexibility.[70] Most computer applications today (for example, word processing) provide the user with an easy-to-use program, but one which is inflexible -- the user must adjust to the conditions set by the application. The Smalltalk approach is to encourage "endless possibilities and easy toolmaking" whereby the user makes his/her own computer tools. Because the burden of making these tools falls upon the user, the system needs to provide easy means to generate tools and also enough pre-written tools to get the user started. Thus, for any given application the user makes a kit of tools (tools for sketching, tools for word processing, etc.), while Smalltalk would provide the "kit maker."

Smalltalk developed a number of marks of style which were original and distinctive. For example, "icons" (tiny pictures), were later adopted in the Lisa and MacIntosh perscoms. In CL applications these icons enlivened text-based material -- they could be used, for example, to animate math problems by making pictorial scenarios of the problem to be solved. Smalltalk emphasized interactive graphics and animation. More than any other system discussed so far, Smalltalk emphasized the active and creative role of the learner. Xerox has not yet benefitted from the commercialization of Dynabook innovations, but Apple Computer Inc. (with their Lisa and MacIntosh computers) has, since they hired Xerox personnel from this project.

5.4.6 Incompatibilities in CL Software

Incompatibilities between different computer systems are a general problem in CL, and are especially a problem for microcomputer software publishers, who need a mass market in order to produce packages in large numbers that can be sold at low cost.

Even between two modules of hardware from the same manufacturer, compatibility may not be possible. In response, a strategy of "upward compatibility" has been developed: system design should ensure that, as one model is superceded by another, software can be passed on to the second system.

One way of solving incompatibilities at the moment is the use of "portability" software for languages and machines that are widely in use. A program is "portable" if it can be cheaply moved from one machine to another. Work on portability software has been done in Canada at both the University of Alberta and University of Waterloo.

Another technical solution to incompatibility is "emulation": making one machine behave as if it were identical to another.

Also, the ways in which CL material is written can minimize the programming portions that need to be modified when adapted for use on a new system.

One level of incompatibility, between two different CL languages, is virtually unbridgeable. Code written in CAN, for example, would have to be completely re-written in NATAL. If a training centre was set up in B.C. using TICCIT and a similar centre was also established in Ontario using CAN, there would be little sharing of courseware between the two.

5.4.7 Content in Computer Learning Applications

Content is the actual learning applications software with which the individual interacts. It is the substance of computer learning. The word "courseware" has been the standard industry term for CL applications in the past, but a distinction has been drawn more recently between "courseware" (entire courses of learning by computer), and smaller chunks of content, called "lessonware".

In the past, as noted earlier, CL content has made only rudimentary use of the computer's capabilities. AI is one major area where advances are occurring; another such area is interaction with the computer by speech. The inability of computers to converse has been a critical communications gap. However, progress is being made in both speech synthesis, and the more difficult speech recognition.

Current speech recognition technology has an extremely limited vocabulary, has problems adjusting to different voices, and has almost no ability to contextualize (to distinguish, for example, between "spirits" as wine and "spirits" as ghosts). Yet its advantages are so enormous that systems capable of recognizing fewer than a hundred words are now commercially available. Speech synthesis is a vastly more simple technical task, and "talking chips" are being used cheaply in a number of consumer products. The first talking consumer product was an educational one: Texas Instruments' "Speak 'n' Spell", which taught spelling.

Despite falling computer prices and technical advances, the cost of developing CL content remain substantial and is a limiting factor. Even with authoring tools, a ratio of 100 hours of development to one hour of content is excellent.

Apart from tools to make content creation faster, other solutions to the high cost of content creation include: (1) an emphasis on merchandising content after it has been developed through sales or licensing; and (2) the use of production centres.

An emphasis on merchandising is part of a more commercial outlook towards content generally in education. The early pattern of teachers with their own short CL programs has been replaced by professional packaged products. In the training world the resale of courseware developed by a firm for its own use is a possibility. There is, however, debate on the likelihood of resale of training material; it is argued that firms' needs are too specialized for custom software to be sold. But in many industries equipment is similar from company to company. Furthermore, there is a general trend in training to delineate sets of "generic skills". Generic training modules with wide relevance can then be used, followed by specialized instruction.

Production centres, either in universities of operated by private firms, can make use of specialized equipment to create CL. They can also bring together production teams.

Several other problems in addition to high costs of content creation are limiting factors for CL software, particularly in the microcomputer marketplace. First, there are computer problems: copies of software programs are easily pirated on floppy discs. Incompatibility between different microcomputers and the poor quality of much CL produced for the school market have been problems as well.

In education, where the use of perscoms suits mass-produced software, content supply problems are poor quality, copyright troubles and incompatibilities. In the training world, with custom-made content dominating, the major problem is the high costs of content creation.

5.4.8 Smart Software in CL

Artificial intelligence programming in CL involves programming that processes ideas and knowledge instead of numbers. AI programs use if-then rules (for example, the computer can be certain that if an animal has tusks, then it is not a cat). It may also use networks of facts that are linked associatively, so that the computer can tell how various pieces of information are related. (The computer can "know," for instance, that both mastodons and elephants have tusks but also "know" that mastodons are extinct.)

After years of basic research, and with advances in computer power, commercial products based on AI are finally appearing. Most are expert systems, computerized consultants equipped with an area of technical expertise. Mycin, a medical advisor, diagnoses certain diseases and prescribes antibiotic drugs. Dipmeter Advisor is a geology expert developed by a major oil service company to interpret data collected from drilling sites. To create Dipmeter Advisor, researchers observed a top field engineer every day for six months, to imitate (through if-then rules) his special expertise, his practical knowledge and experience, and his means of problem-solving. An AI system will also explain how a conclusion was reached, so that human experts can assess the system's judgment. While there are only a few hundred expert systems in use today, a flood of others are expected, beginning in specialized scientific and industrial settings, and spreading to factories and offices and to consumer markets, perhaps by the end of the decade.

The limitations at present are the amount of time needed to create the "expert", good interfaces and the power of computer resources required. Already, however, a new generation of microcomputers can use Prolog and LISP (the main AI programming languages).

By nature, educational computing calls for the use of knowledge-based computers and, as noted earlier, a number of early examples of AI-based systems were produced for CL. SOPHIE, for example (developed at Xerox PARC), is an expert in troubleshooting electronic circuits. It can interact with a student in natural language, and either SOPHIE or the student can ask questions. As the student practices with the simulated circuit, SOPHIE knows what he or she should be trying to do: if a mistake is made, some pointed questions will be asked. BUGGY is an expert in the kinds of errors ("bugs") students make in doing simple math problems. WHY, an expert in meteorology, takes the Socratic approach to tutoring: if a student overgeneralizes about a cause of weather conditions, WHY raises a counter-example.

While hardware advances have proceeded at breakneck speed, the quality and sophistication of educational software has lagged behind. The lack of quality content is the problem most often cited by teachers using perscoms in schools. In the future, AI-based approaches can be expected to dramatically improve CL.

5.4.9 Videodisc Systems

A considerable amount of educational/training software is being written for videodiscs driven by personal computers. A videodisc system consists of a videodisc player, a tv set, and the disc itself, which acts as a high-capacity storage medium for video information with two audio channels. Each side of the disc can hold 54,000 tv "frames."

The use of videodisc for education and training is at an early stage:[71] but in industrial settings, "multimedia" training is becoming a catchword, and a number of large organizations have embraced videodisc technology. Most major CL systems have developed video interfaces. In the educational world, experimental projects have explored the use of microcomputer-controlled videodiscs for interactive learning. Eventually, the marriage of a low-cost microcomputer unit with a low-cost videodisc player may produce a medium for CL that is cheap, powerfully interactive, and capable of full-video TV.[72]

A computer learning application using videodisc can mix video segments (such as narrative sequences, to add interest), video illustrations, graphic illustrations, or printed text. Each frame can be held as long as the individual wants to view it. Thus 20 minutes of videodisc "running time" (about 35,000 frames) could equal six to twelve hours of actual computer learning time.

Three technologies were developed for videodisc systems: the optical or laser system, and two types of "capacitance" systems--all incompatible. There are also three general types of interactivity offered by videodisc systems, depending on the hardware used. Consumer videodisc units (such as those sold for home entertainment) allow a very limited amount of user control -- such as jumping to a particular "frame". More sophisticated "institutional" or "educational" players have been used for CL, and retail at higher prices ($2,000 to $3,000 US). These models have microprocessors that can carry out simple programming functions for greater capabilities. A videodisc player can be interfaced to a microcomputer.

Using a microcomputer to control a videodisc player allows full-scale computer programs to be written to control the presentation of material and to store and analyze student response data. It also allows for computer-generated text overlays and graphics, for full alphanumeric input, and adaptive branching (based on the learner's cumulative performance).

The costs of producing content on videodisc can vary widely but will reflect both the high cost of CL plus the high price of video production. Very different types of video sequences could be used, i.e. studio vs. location. It could cost from $2500 to $25,000 to make a disc, and the best rule of thumb at present allows $1000 to $10,000 per videodisc minute. (Ten minutes, however, could be equivalent to several hours of instruction using stillframes).

There are also costs to produce the master from which copies are made. This process costs from $3000 to $5000. Actual reproduction of the discs themselves is inexpensive and gets lower and lower as more copies are made.

5.4.10 The Educational Market

The following sections discuss three potential markets for computer learning software: education (in schools and post-secondary institutions); training, where industry or government trains the workforce; and the consumer marketplace.

The markets for computer learning are not well understood. Training, for example, is expected to provide the most opportunity for CL in the next few years. Yet there is scant information about industry expenditures on training via traditional means. The consumer market is the biggest question mark--the most difficult to predict, and possibly the largest in dollars over time. CL in the personal computer market is like a newly discovered species; it is difficult to even attach a name. Free-form types of learning such as LOGO's turtle geometry are closer to play than to didactic teaching, and sudden fads may well occur.

Education has been a maternal marketplace for computer learning, providing settings for research and development at universities, and sparse but sustaining use in schools and higher education through CL's early years.

Annual spending on education represents approximately 8% of Gross National Expenditures, almost all of it supported by government. Over $26 billion was spent on education in Canada in 1982.[73] Two-thirds of this money goes to elementary/secondary schools, and nearly $7 billion is spent on colleges and universities (Exhibit 5-8). Yet despite its attractive size and the prospect of policies by provincial governments that could aid Canadian suppliers, the educational market is problematic.

The use of microcomputers in schools has grown dramatically. In the US the number of units in schools went from 40,000 in 1980 to 291,000 by mid-1983, with predictions of 2 million by 1988. In Canada, by 1983 33,000 units were in schools, and 27,000 more were estimated to be sold in 1984.[74]

Generally three incompatible systems dominate in schools both in Canada and the US. Apple, Tandy Radio Shack, and Commodore. All three are US manufacturers with divisions in Canada. However, the Icon, the new Canadian educational microcomputer of Cemcorp of Toronto, is unique not only because of Canadian manufacture, but also because it is designed specifically for educational use. It can also run both 16 and 32 bit generations of software.

The incompatibilities in this market make it a difficult one for a would-be software publisher.

EXHIBIT 5-8

Expenditures on Education in Canada
(Elementary/Secondary, Community Colleges and Universities)
($M)

	1979-80	1980-81	1981-82	1982-83
Elementary/Secondary	13,399	15,192	17,254	19,000
Post Secondary:				
Community Colleges	1,611	1,796	2,026	2,225
Universities	3,948	4,360	4,827	5,319
Total	$18,958	$21,349	$24,108	$26,544

Source: Statistics Canada, Catalogue No. 81-220, Advance Statistics in Education, 1981-1982, p.25.

CL software is the centre of the educational market and the lack of good material is evident in both Canada the US, and Asia. Many hardware purchase decisions are swayed by the availability of good material for a specified brand of micro-computer. Pressure is being exerted upon provincial governments by teachers to organize content supply through clearinghouses, and user groups are making efforts to share content and set up libraries. The shortage of applications is a major problem.

This would seem to be a seller's market. Furthermore, Canadian content suppliers could look forward to the kind of preferential treatment that has made textbooks the only category of books in Canada where more is spent for Canadian products than for imports.[75] But content suppliers face several problems, including the tendency among teachers to copy diskettes and the incompatibilities that sever the market.[76]

Thus, from an industry viewpoint, the school market for CL in Canada, despite the rapid growth of microcomputers, has been viewed as more uncertain than the training market.

5.4.11 The Training Market

Training is a major activity in Canada and costs government and industry near $4B a year, including manpower and other training supported by government. It also includes training carried out in industry, estimated at $600 million annually, and approximately $600 million per year for training within the military. It is widely expected that massive retraining efforts will be required as computerization changes jobs and causes employment shifts.

This figure, however, includes not only expenditures for training but also salaries and wages for trainees, which vary but typically comprise about 50% of total training costs. A further problem in estimating the training market is that scarcely any information is available about training carried out in industry. Moreover, the information that exists is complicated by overlaps between government-supported training and training supported by industry itself. Finally, it is difficult to estimate the degree to which computer-based training (CBT) could replace standard methods. It has been suggested, for example, that 5% to 50% of the millions spent on training in the military could involve computers--an estimate that allows for a wide margin of uncertainty.[77]

Much greater amounts are spent annually on education than on training, but marketplace conditions for the latter are more attractive, and training is one of the most opportune markets for computer learning software today.

A typical pattern for computer-based training consists of a major purchase of hardware and authoring software, using systems like CAN and TICCIT, followed by a substantial course development effort. While there may be uncertainty about the total potential training market, large organizations find computer-based training cost-effective, typically reducing training time by 30%.

Computer-based training in industry makes economic sense. B.C. Tel., for example, has large training needs and spends $12M a year training its employees. One of its most popular courses is Digital Logic, a basic course taken by employees in dealing with computers throughout B.C. Tel operations. Over 250 students take this course at a cost to the company of $1M a year. Converting the course to computer form reduced the time needed to train from ten days to five days. Course development costs amounted to $175,000, and equipment costs for a TICCIT system that supports 20 terminals were $500,000. Thus at least $375,000 is saved per year (more if students complete courses quicker), and the system will pay for itself within three years.[78]

B.C. Tel's digital logic course is also an example of "generic" training that has a market outside of the company that created it. Telephone companies worldwide use similar technology, and the B.C. Tel Education Centre has been selling both its training course and the TICCIT system itself on a dometic and an international basis.

There are many other examples of major uses of CBT. Pilots at United Airlines train on a flight simulator (supplied by CAE Electronics Ltd. of Montreal) that is so real they sometimes emerge shaking from a simulated head-on collision. The simulator costs some $10M but is still only a fraction of the cost of a real airplane and can be operated at one-tenth the cost.[79] IBM has used computer simulation in its executive management training program for a number of years.[80] Digital Equipment Corporation distributes computer-based instructional packages to its training centres for management instructions, technical training and customer education. Holiday Inns, Inc. uses computer simulations of hotel operations to train managers in business planning and financial forecasting skills. There have been experiments with the use of computer simulations for nursing in Quebec.[81] The large Credit Agricole bank in France has used micro PLATO for self-paced training; computer-based training is being used in nuclear power plants in France, and the list is expanding.

Most CBT examples discussed here exemplify the mini-computer or mainframe model of training, where multiple users access central computer resources. A new trend is also beginning as microcomputers diffuse in small businesses and within large corporations. It was estimated (in 1981) that some 750,000 business persons were using microcomputers.[82] One recent survey of 160 large firms found that 50% of the companies responding were using computers to train, and that 71% of these included the use of microcomputers.[83] While the capacity of microcomputers is still limited (and simulations, for example, might exceed microcomputer capacity), an increasing use of 32 bit microcomputers can be expected, and eventually good packaged software products will evolve.

The federal government supports training in industry through several programs that contribute to costs of training and trainee wages. It also supports the large Manpower Training Program, which buys seats in classes for trainees and provides financial support through unemployment insurance. The federal government spends $1 billion per year in training. But although it supports an enormous amount of classroom training through its Manpower Training Program, because of a federal/provincial split in jurisdiction, it has no control over curriculm development. It would require considerable arranging, for example, for the Manpower Program to instigate the use of computers in training.

Another type of informal and invisible training comprises a certain market as well outside of industry and government, especially for training in the use of new computer equipment. People have to learn to use new equipment and require instruction to do so. Short courses are being offered by private vocational schools, by computer stores and by companies who supply computer services. This applications area is especially suited to CL. Some manufacturers of large hardware systems are including training software packages with hardware purchases, and independent entrepreneurs are marketing software packages teaching how to program microcomputers.

5.4.12 The Consumer Market

The "consumer market" for computer learning consists of learning outside of either educational institutions or organizations which train employees. Computer learning in this market is associated with personal computers.

Sales of perscoms have been multiplying for several years and show the most rapid growth in the computer industry. It is estimated that the home computer market is growing by 100% annually and that 14 million US homes will have computers by 1987.

At least three types of computer learning suggest themselves for the consumer marketplace: 1) packaged software for training in the use of microcomputers; 2) informal instruction, both for small businesses and in the home; and 3) free-form unstructured learning for children. Rapid growth in "educational software" has occurred in this last area in the past few years. Annual sales of educational software have been widely predicted to soar to $3 billion within five years.

5.4.13 The Cemcorp Icon

A basic decision in planning for CL is whether to "make or buy" software and equipment. Ontario is trying to promote Canadian capabilities in both hardware and software with a new educational perscom, the Icon. The Icon, can run both 16 bit and 32 bit generations of software. Also, to minimize the software lag, a summer project at OISE focused on conversion of software which is written in Basic into the Canadian version, Waterloo Microbasic, for 20 to 30 of the better educational software packages. Based on a Unix-like system, Icon operates by manipulating icons which represent computer functions. Ontario has recently made

$5 million available for the development of Icon software, to rise to $10 million next year.

Other provinces have taken the opposite approach, (buying Apples and subscribing to US content. The western provinces have emphasized Apple-derived systems (i.e. either Apple or Bell & Howell units)

Ontario's plan for the Canadian Educational Microcomputer is based on a subsidy to schools, which receive a larger provincial grant for equipment if they purchase an Icon.

If Ontario becomes a leader in CL, will other provinces cooperate? Quebec is on its own because of language — it will have little interest in sharing clearinghouses with other provinces, for example, and will have to develop its own French clearinghouse and its own content. However, it could be to Quebec's advantage to translate some of the educational software being developed in Ontario. Also, the development of microcomputers in education has been slow in Quebec and a great brouhaha has been recently created, as Quebec announced it had bought an educational perscom from France of which the first version was widely reported to be a technological lemon.

In English-speaking provinces, will there be a tendency to support a Canadian hardware product, or use Canadian material? Because different microcomputers are incompatible, content is a major influence in a purchasing choice. The decision to buy an Icon may amount to a comparison between Icon content and existing US content.

The Ministry of Education of Ontario has given Cemcorp a short term monopoly on manufacturing these educational microcomputers, but any micro which meets the above specifications and which has 60% Canadian content in its hardware is eligible for the Ministry's 75% subsidy for purchases by school boards. Cemcorp, under the leadership of Meridian Technologies, now has purchase agreements with Ontario's Ministry of Education to buy Icons. The 16 bit Icon built on an Intel 80186 chip has 128K RAM, which is expandable to 250K. Its RAM is large enough to support a full screen text editor, user interfaces, graphics and networking capabilities. This basic student microcomputer can run a diversity of languages, including Waterloo Microbasic, Logo or Miro Pascal. An advanced 32 bit microprocessor has 256K RAM, which is expandable to 1024K, with voice synthesizer and tone generator capabilities, a number of display formats, Telidon graphics capabilities and several programming languages including C, Micro APL, Waterloo Micro Cobal and Micro Fortran.

With respect to software for the Icon, the Ministry has required 95% Canadian content for systems and utilities software, with no such rules pertaining to applications software.

EXHIBIT 5-9

Typical Instances of Computer Learning

The Military

In 1981, a flurry of interest erupted in computer circles when Canada purchased 137 F-18 fighter planes from McDonnell Douglas. 1500 hours of computer-based instruction for training for these planes was required, which would have meant an infusion of $25M to $50M to a Canadian software firm. The outcome, however, was disappointing: a division of McDonnell Douglas in the US received the training contract. Historically the military in Canada has not played an "engine" role in CL development.

Training in the Federal Pubic Service

The cost of career training activities attended by employees in the federal public service in 1980-1981 was $153.6M.* This includes $70M for participants' salaries, and was a 39% increase over 1979-1980 spending. 22% more individuals underwent training in 1980-1981. Most training is provided by departments themselves, and "technical and professional" categories account for over half of all training courses.

In addition, the Public Service Commission spends over $29M a year in official language training.** In 1981 there were over 2300 public servants studying French for four hours a day, while 659 people were studying English.

* Treasury Board of Canada Secretariat (1982). This figure comprises the major expenditure on training. There is in addition language training and several small programs such as "educational leave."

** Public Service Commission of Canada (1981), pp. 40, 109.

* * * * *

With this background, we now turn to an examination of the software industries and respective policy support in five countries -- Canada, Brazil, Singapore, the US and Japan.

6.0 SOFTWARE INDUSTRIES AND SUPPORT IN FIVE COUNTRIES

6.1 Canada

In a 1983 study done by the Evans Research Corporation,[84] relying mainly on 1981 data, the following empirical picture of the Canadian software industry emerged:

1. The Canadian domestic market for computer software was $608 million in 1981, an increase from $457 million in 1980. Evans anticipates that this market will grow at an annual rate of 28%, reaching $5.4 billion by the end of the current decade. Applications software presently is the fastest growing market segment, but the 1980 and 1981 markets for applications software were only $114 million and $161 million, respectively. Evans anticipates that the applications software market will be $2.2 billion by 1990, showing an annual average growth increase of 34% in the 1981 to 1990 period. Sales of custom software, on the other hand, which was $215 million in 1980 (47% of the total software market), will increase to only $1.4 billion by 1990, or 25% of the total software market.

2. In 1983 there were over 1,000 firms located in Canada supplying custom or applications software to the Canadian market. However, revenues from 516 of the largest software suppliers which were captured in this study accounted for 98% of this total market.

3. In 1981, 53 suppliers had software revenues which exceeded $1 million. Of these 53 suppliers, 28 were foreign owned, and the revenues of these 28 foreign owned companies comprised 54.3% of the total 1981 Canadian software market. The other twenty-five Canadian owned software suppliers created revenues which comprised 23% of the 1981 software market.

   Of these 53 key software suppliers, 24 were manufacturers of hardware. In 1981, the software revenues of these hardware manufac turers comprised \$326.4 million, or 54% of the total Canadian software market.

It is anticipated that by 1990 however that the total revenues from Canadian software houses and EDP consultant firms will have grown to 42% of the total Canadian market, while the share of the hardware manufacturers will decline to 38% of this market. Evans has predicted that by 1986, in both Canada and the US, revenues of independent suppliers of software will exceed the software revenues of the hardware suppliers. This change will be dominated by applications software for personal computers.

4. More than 4,000 people are presently employed in some phase of software development in Canada.

5. Evans and others have discerned a number of trends which characterized the Canadian software industry; the use of (pre-written) packaged software is increasing, while the use of custom software is rapidly diminishing. Secondly, software is becoming a crucial factor in any buyers' decision to purchase a particular piece of computer hardware. In other words, software is becoming a marketing tool utilized by the vendors to sell their equipment. Finally, there is a trend of mergers and acquisitions in the (business) software industry.

The Evans study considered that there were several obstacles to the success of Canadian software; intense competition from US firms, negative Canadian perceptions of domestic software products and the small Canadian market size. In addition, the report noted that although both federal and provincial Canadian governments have recently enacted programs directed toward high technology industries in general, there are none that are software-specific, designed to help a growing Canadian software industry. The report also noted that there is not presently any coordinated approach by various levels of govermment to develop Canadian software.

The report finally summarized comments from sixteen software firms concerning the question of whether the government should focus in the area of grants or tax/fiscal aids. The only companies to support grants and subsidies in this survey were those already significantly subsidized by the government.

6.1.1 Detailed View

The 53 firms identified by Evans as having 1981 revenues of $1 million or greater are listed in Exhibits 6-1 and 6-2 in a ranking by their 1981 total software revenues as a percent of the firm's total EDP revenues.

In both 1980 and 1981, these (mainly foreign) firms accounted for more than 77% of those years' respective markets. Software revenue growth during these two years for this group was more than 33% annually.

Evans' second category, the 463 firms with average software revenues of less than $1 million, is composed mainly of cottage industries and small businesses. These generated 1980 revenues of $138.29 million and accounted for 22.7% of the 1981 market.

Exhibit 6-3 presents an estimation of software revenues of these 53 key software suppliers, by vendor type. This group was dominated by 24 hardware manufacturers, and these hardware vendors generated 54% of total 1981 software revenues.

Exhibit 6-4 presents the geographic distribution of the 53 key Canadian software suppliers, and Exhibit 6-5 presents Canadian software suppliers by country of ownership. Fifty-seven percent (30) of the 53 key suppliers were located in Metro Toronto.

EXHIBIT 6-1

Top Companies in the Canadian Information Processing Industry

(Ranked by 1981 Software Revenues)

Company	1981 Ranking	1981 Software Revenues	1980 Ranking	1980 Software Revenues
IBM Canada ltd.	1	182.1	1	145.9
Digital Equipment (Can)	2	37.9	2	24.6
Systemhouse Ltd.	3	26.0	3	16.2
Canada Systems Group	4	20.3	7	14.0
BC Systems Corp.	5	19.7	4	18.6
Control Data Canada	6	18.1	5	16.3
NCR Canada Ltd.	7	15.2	6	15.0
Sperry Inc.	8	12.7	8	10.6
Burroughs Inc.	9	11.4	9	9.4
DMR & Assoc. Ltd.	10	11.2	10	7.1
Quasar Systems Ltd.	11	10.7	12	5.4
Hewlett-Packard	12	8.9	11	7.0
GEAC Computer Ltd.	13	7.0	13	4.7
Bailey & Rose Ltd.	14	5.0	17	3.0
IP Sharp Assoc. Ltd.	15	4.4	18	2.6
Computel Systems Ltd.	16	4.1	15	3.8
ISS Info. Sys. Serv.	17	3.52	21	2.38
Computech Cons. Can.	18	3.9	19	2.5
MSA Canada Inc.	19	3.1	16	3.2
AES Data Ltd.	20	3.1	25	1.8
Tandy (Radio Shack)	21	3.0	-	N/A
STS Systems Ltd.	22	3.0	20	2.5
Data General (Can)	23	2.9	11	2.3
Philips Info Systems	24	2.8	24	2.0
TRW Canada (Datapoint)	25	2.8	22	2.2
Cincom (Can) Ltd.	26	2.8	23	2.2
Cullinane Canada Ltd.	27	2.6	53	.5
Xerox Canada Inc.	28	2.3	26	1.7
Wang Canada Ltd.	29	2.2	39	1.1
Mohawk Data Sci.	30	2.2	27	1.7
Nfld. & Lab. Comp. Ser.	31	2.1	28	1.6
Sydney Dev. Corp.	32	2.1	75	.23
Apple Canada Ltd.	33	1.9	34	1.2
Act Comp. Ser. Ltd.	34	1.9	45	.8
University Comp.	35	1.9	32	1.3
Honeywell Ltd.	36	1.8	31	1.4
CTS Computer Sys.	37	1.8	35	1.2
Le Group Bst Ltd.	38	1.8	29	1.5
Cableshare Inc.	39	1.7	40	1.1

EXHIBIT 6-1
(continued)

Company	1981 Ranking	1981 Software Revenues	1980 Ranking	1980 Software Revenues
CGE (Info. Serv.)	40	1.6	33	1.3
Tandem Comp. Can.	41	1.5	54	.5
ICL Canada Ltd.	43	1.5	36	1.2
Tyme Systems Ltd.	44	1.5	46	.8
Four Phase Sys. Ltd.	45	1.3	37	1.2
Dyad Computer Sys.	46	1.26	43	1.02
IST Inc.	47	1.2	42	1.0
Prime Computer (Can)	48	1.2	41	1.1
Electrohome Ltd.	49	1.2	44	.9
Nabu Manu. Corp.	50	1.1	-	N/A
Computrex Centres	51	1.1	38	1.2
Omnitech Graphics	52	1.1	88	.05
ADP Inc.	53	1.0	47	.8
Ahearn & Soper Inc.	54	.9	48	.7
Nelma Data Corp.	55	.8	59	.4
Northern Telecom	56	.792	49	.63
Perki-Elmer (Can)	57	.7	50	.6
Real Time Datapro	58	.64	52	.51
STC Canada Inc.	59	.6	66	.3
Saskcomp	60	.6	60	.4
Computer Sc. Can.	61	.6	51	.6
Commodore Bus. Mach.	62	.6	61	.4
Olivetti Can. Ltd.	63	.5	55	.5
Texas Instruments	64	.5	56	.5
Dataline Inc.	65	.5	62	.4
Centronics Can. Inc.	66	.5	57	.5
Digitech Ltd.	67	.5	63	.4
Nixdorf Canada Ltd.	68	.5	76	.2
Micos Inc.	69	.5	67	.3
Maritime Computers	70	.45	64	.35
Polycom Systems Ltd.	71	.45	65	.33
Boeing Comp. Serv.	72	.42	58	.42
Tektronix Can. Inc.	73	.4	77	.2
Automation Cen.	74	.4	68	.3
G.A. Computer Ltd.	75	.4	69	.3
Norpak Ltd.	76	.35	74	.25
Amdahl Ltd.	77	.3	84	.1
Lanpar Tech. Inc.	78	.3	70	.3
Cons. Comp. Inc.	79	.3	78	.2
Data Terminal Mart	80	.3	79	.2
Nat. Datacentres	81	.3	71	.3
Greyhound Comp.	82	.3	72	.3

EXHIBIT 6-1
(continued)

Company	1981 Ranking	1981 Software Revenues	1980 Ranking	1980 Software Revenues
Plessey Per. Sys.	83	.3	80	.2
D.E. McMullen & Assoc.	84	.26	73	.26
Anderson Jacobson	85	.25	82	.18
ITT Canada Ltd.	86	.2	85	.1
Recognition Equipment	87	.2	86	.1
Volker-Craig Ltd.	88	.2	87	.1
Digital Bus. Comp.	89	.2	81	.2
Norango Comp. Sys.	90	.18	83	.15
ESE Limited	91	.1	89	.05
KO Mair & Assoc.	92	.1	90	.05
Matrox Elect. Sys.	93	.09	91	.05
Comcheq Serv. Ltd.	94	.08	92	.05
Coverall Comp. Serv.	95	.08	93	.05
Cybernex Ltd.	96	.078	94	.05
Comtech Group Inter.	97	.05	95	.05
Riley's Datashare	98	.02	96	.01
Datacrown Inc.	99	0.0	97	0.0
MAI Canada Ltd.	100	0.0	98	0.0
Memorex Canada Ltd.	101	0.0	99	0.0
Gandalf Tech. Inc.	102	0.0	100	0.0
Cherney-Mills Inc.	103	0.0	101	0.0
Manitoba Data Serv.	104	0.0	102	0.0
General Datacomm.	105	0.0	103	0.0
Comshare Ltd.	106	0.0	104	0.0
Comterm Inc.	107	0.0	105	0.0
Telex-Tulsa Comp.	108	0.0	106	0.0
Zentronics Div.	109	0.0	107	0.0
Develcon Elect. Ltd.	110	0.0	108	0.0
Dasco Data Products	111	0.0	109	0.0
Datamex Ltd.	112	0.0	110	0.0
NCR Comten Inc.	113	0.0	111	0.0
Cybershare Ltd.	114	0.0	112	0.0
MICR Systems Ltd.	115	0.0	113	0.0
DP Consultants	116	0.0	114	0.0
Zavitz Electronics	117	0.0	115	0.0
Boothe Comp. Ltd.	118	0.0	116	0.0
Computer Ut. Man.	119	0.0	117	0.0
Altel Data	120	0.0	118	0.0
Alphatext Inc.	121	0.0	119	0.0
Datatech Systems	-		-	
TOTAL		486.77		365.72

Source: Evans Research Corporation

EXHIBIT 6-2

Top Companies in the Canadian Information Processing Industry

(Ranked by Software Revenues as a Percentage of the Company's Total EDP Revenues)

Company	1981 Rank	% of 1981 EDP Revenues	Total 1981 EDP Revenues (millions)	1980 Rank	% of 1980 EDP Revenues	Total 1980 EDP Revenues (millions)
Cullinane Can.	1	100.0	2.6	1	100.0	.5
Bailey & Rose	2	100.0	5.0	2	100.0	3.0
Computech Cons.	3	100.0	3.9	3	100.0	2.5
MSA Canada Inc.	4	100.0	3.1	4	100.0	3.2
Cincom Systems Can.	5	100.0	2.8	5	100.0	2.2
University Comp.	6	100.0	1.9	6	100.0	1.3
Quasar Systems Ltd.	7	96.4	11.1	7	94.7	5.7
ISS Info. Systems Serv.	8	95.0	3.7	8	95.0	2.5
Systemhouse Ltd.	9	85.3	30.5	9	32.9	19.3
DMR & Assoc. Ltd.	10	60.0	18.7	10	60.0	11.9
Dyad Computer Sytems	11	60.0	2.1	11	60.0	1.7
Le Group Bst. Ltd.	12	50.0	3.5	12	50.0	3.0
Sydney Dev. Corp.	13	42.9	4.9	13	38.8	.6
Omnitech Graphics	14	39.2	2.6	23	16.0	.3
BC Systems Corp.	15	34.0	57.9	14	37.9	49.1
STS Systems Ltd.	16	33.3	9.0	15	36.8	6.8
Computrex Centr.	17	31.0	3.5	16	31.6	3.8
Nfld. & Lab. Comp.	18	25.0	8.4	19	21.9	7.5
Cableshare Inc.	19	25.0	6.6	17	25.0	4.2
Tyme Systems Ltd.	20	25.0	5.9	20	20.0	4.2
CTS Computer Systems	21	25.0	7.0	18	25.0	4.7
ACT Comp. Serv.	22	24.0	7.5	26	11.0	6.8
Canada Systems Group	23	20.0	101.5	21	20.0	77.9
GEAC Comp. Ltd.	24	20.0	34.2	22	20.0	23.6
Nabu Manu. Corp.	25	15.7	7.0	-	-	N/A
Digital Equipment	26	15.0	252.6	24	15.0	163.7
Plessey Per. Systems	27	15.0	2.2	28	10.0	1.8
IBM Canada Ltd.	28	12.0	1,517.7	25	12.0	1,216.5
GA Computer Ltd.	29	12.0	3.7	29	10.0	2.7
Hewlett-Packard	30	11.0	81.0	27	11.0	63.5
Control Data Can.	31	10.0	181.3	30	10.0	162.6
NCR Canada Ltd.	32	10.0	152.4	31	10.0	150.1
Burroughs Inc.	33	10.0	114.0	32	10.0	94.0
IP Sharp Assoc.	34	10.0	42.0	57	7.0	35.3
Computel Systems Ltd.	35	10.0	41.3	33	10.0	38.4
Data General (Can)	36	10.0	29.0	34	10.0	23.0
TRW (Datapoint)	37	10.0	28.0	35	10.0	22.0
MDS Canada Ltd.	38	10.0	22.3	36	10.0	16.5

EXHIBIT 6-2
(continued)

Company	1981 Rank	% of 1981 EDP Revenues	Total 1981 EDP Revenues (millions)	1980 Rank	% of 1980 EDP Revenues	Total 1980 EDP Revenues (millions)
Apple Can. Ltd.	39	10.0	20.1	-	-	0.0
Tandem Comp. Can.	40	10.0	15.7	37	10.0	5.2
ICL Canada Ltd.	41	10.0	15.0	38	10.0	15.3
Four Phase Systems	42	10.0	13.4	39	10.0	12.4
Prime Computer	43	10.0	13.0	40	10.0	10.0
Electrohome Ltd.	44	10.0	12.0	41	10.0	9.2
Nelma Data Corp.	45	10.0	7.6	42	10.0	4.2
Perkin-Elmer (Can)	46	10.0	6.8	43	10.0	6.1
Real Time Datapro	47	10.0	6.4	44	10.0	5.1
Micos Inc.	48	10.0	5.0	45	10.0	3.2
Maritime Comp.	49	10.0	4.5	46	10.0	3.5
Polycom Systems Ltd.	50	10.0	4.5	47	10.0	3.3
Boeing Comp. Ser.	51	10.0	4.2	48	10.0	4.2
Automation Cen.	52	10.0	4.0	49	10.0	3.0
DE McMullen	53	10.0	2.6	50	10.0	2.6
Anderson Jacob.	54	10.0	2.5	51	10.0	1.8
Digital Bus. Com.	55	10.0	2.2	52	10.0	2.2
Norango Com. Systems	56	10.0	1.8	53	10.0	1.5
Sperry Inc.	57	9.0	141.0	54	8.0	132.0
Nixdorf Canada	58	8.0	5.8	55	8.0	2.9
Tandy (Radio Shack)	59	7.5	39.7	-	-	22.3
Ahearn & Soper	60	6.4	14.0	58	5.5	12.8
Centronics Can.	61	5.5	9.1	56	7.4	6.8
Xerox Canada	62	5.0	45.2	59	5.0	33.9
Wang Canada Ltd.	63	5.0	44.5	60	5.0	21.0
CGE (Info. Serv.)	64	5.0	31.0	61	5.0	26.0
IST Inc.	65	5.0	23.8	62	5.0	20.4
ADP Inc.	66	5.0	20.0	63	5.0	16.7
Computer Sc. Can.	67	5.0	12.2	64	5.0	11.2
Commodore Bus.	68	5.0	11.9	65	5.0	7.4
Texas Instruments	69	5.0	10.0	66	5.0	10.8
Dataline Inc.	70	5.0	9.9	67	5.0	8.2
Digitech Ltd.	71	5.0	9.0	68	5.0	7.9
Norpak Ltd.	72	5.0	7.0	69	5.0	5.0
National Datacentre	73	5.0	6.3	70	5.0	5.2
Greyhound Comp.	74	5.0	5.9	71	5.0	5.1
Saskcomp	75	4.0	16.8	73	3.3	12.8
Tektronix Can.	76	4.0	9.0	72	4.0	6.0
Northern Telecom	77	3.0	26.4	74	3.0	21.0
Data Ter. Mart	78	3.0	8.5	75	2.0	7.5
KO Mair & Assoc.	79	2.44	4.1	86	1.52	3.3
Honeywell Ltd.	80	2.1	86.0	85	1.9	68.6
Philips Info.	81	2.0	140.0	76	2.0	100.3

EXHIBIT 6-2
(continued)

Company	1981 Rank	% of 1981 EDP Revenues	Total 1981 EDP Revenues (millions)	1980 Rank	% of 1980 EDP Revenues	Total 1980 EDP Revenues (millions)
Olivetti Can.	82	2.0	26.5	77	2.0	24.0
Lanpar Tech.	83	2.0	16.4	78	2.0	15.0
Cons. Comp. Inc.	84	2.0	15.3	79	2.0	11.9
ITT Canada Ltd.	85	2.0	9.5	80	2.0	7.2
Recognition Eq.	86	2.0	8.1	81	2.0	6.3
Volker-Craig	87	2.0	7.6	82	2.0	6.3
Comcheq Serv.	88	2.0	4.8	83	2.0	3.6
Coverall Comp.	89	2.0	4.2	84	2.0	2.6
AES Data Ltd.	90	1.8	172.8	88	1.2	155.4
Cybernex Ltd.	91	1.5	5.2	87	1.5	3.1
Amdahl Ltd.	92	1.1	72.6	94	.58	43.0
STC Canada Ltd.	93	1.0	56.4	89	1.0	28.8
NAS Canada ltd.	94	1.0	15.0	90	1.0	12.0
ESE Limited	95	1.0	10.0	91	1.0	5.4
Matrox Elect.	96	1.0	8.7	92	1.0	4.7
Comtech Gr. Int.	97	1.0	5.8	93	1.0	5.7
Riley's Datashare	98	.5	5.6	95	.5	5.1
Datacrown Inc.	99	0.0	86.2	96	0.0	68.6
MAI Canada ltd.	100	0.0	47.1	97	0.0	39.4
Memorex Canada	101	0.0	40.5	98	0.0	30.8
Gandalf Tech.	102	0.0	40.2	99	0.0	26.2
Cherney-Mills	103	0.0	15.9	100	0.0	20.7
Man. Data Serv.	104	0.0	15.5	101	0.0	13.0
General Dataco.	105	0.0	13.9	102	0.0	12.5
Comshare Ltd.	106	0.0	10.2	103	0.0	7.6
Comterm Inc.	107	0.0	9.5	104	0.0	6.0
Telex-Tulsa	108	0.0	9.0	105	0.0	6.0
Zentronics Div.	109	0.0	8.0	106	0.0	4.0
Dasco Data Prod.	110	0.0	6.2	107	0.0	5.1
Datamex Ltd.	101	0.0	5.7	108	0.0	4.8
NCR Comten Inc.	112	0.0	5.2	109	0.0	2.5
Cybershare Ltd.	113	0.0	4.6	110	0.0	4.3
MICR Systems	114	0.0	4.2	111	0.0	4.2
DP Consultants	115	0.0	3.6	112	0.0	3.0
Zavitz Elect.	116	0.0	3.2	113	0.0	2.6
Boothe Comp. Ltd.	117	0.0	3.0	114	0.0	2.9
Computer Ut. Man.	118	0.0	2.6	115	0.0	2.1
Altel Data	119	0.0	15.0	116	0.0	12.0
General Datacom	120	0.0	13.9	117	0.0	12.5
Alphatext Inc.	121	0.0	4.9	118	0.0	4.7
Datatech Systems	-		6.5	-		4.8

Source: Evans Research Corporation

EXHIBIT 6-3

ESTIMATED SOFTWARE REVENUES OF THE 53 KEY SOFTWARE SUPPLIERS BY VENDOR TYPE IN THE CANADIAN SOFTWARE INDUSTRY

Vendor Type	No. of Firms	Total 1981 Software Revenues ($ Millions)	Total 1980 Software Revenues ($ Millions)	Percentage Growth 1980-1981
HARDWARE MANUFACTURERS	24	326.4	254.4	28.3
Mainframe Manufacturers	6	227.6	184.8	23.2
Minicomputer Manufacturers	11	82.4	61.8	33.3
Microcomputer Manufacturers	3	6.0	1.2	400.0
Word Processor Manufacturers	4	10.4	6.6	57.6
SERVICE BUREAUX	11	59.4	46.8	26.9
SOFTWARE HOUSES AND EDP CONSULTING FIRMS	11	72.18	43.73	65.1
OEM'S SYSTEMHOUSES, DISTRIBUTORS AND THIRD PARTY COMPANIES	5	9.2	6.05	52.1
TERMINAL AND PERIPHERAL EQUIPMENT SPECIALISTS	2	2.8	2.2	27.3
TOTAL	53	469.98	353.18	33.1
GRAND TOTAL		608.27	456.83	

Source: Evans Research Corporation

EXHIBIT 6-3
(continued)

Vendor Type	Percentage of Total 1981 Software Revenues Generated by the Key 53 Software Suppliers	Percentage of Total 1980 Software Revenues Generated by the Key 53 Software Suppliers	Percentage of Total 1981 Canadian Software Market	Percentage of Total 1980 Canadian Software Market
HARDWARE MANUFACTURERS	69.4	72.0	54.1	56.5
Mainframe Manufacturers	48.4	52.3	37.7	41.1
Minicomputer Manufacturers	17.5	17.5	13.7	13.7
Microcomputer Manufacturers	1.28	.3	1.0	.3
Word Processor Manufacturers	2.2	1.9	1.7	1.5
SERVICE BUREAUX	12.6	13.3	9.8	10.4
SOFTWARE HOUSES AND EDP CONSULTING FIRMS	15.4	12.4	12.0	9.7
… HOUSES, DISTRIBUTORS AND THIRD PARTY COMPANIES	2.0	1.7	1.5	1.34
TERMINAL AND PERIPHERAL EQUIPMENT SPECIALISTS	.6	.6	.5	.5
TOTAL			77.9	78.4

Source: Evans Research Corporation

EXHIBIT 6-4

GEORGAPHIC DISTRIBUTION OF 53 KEY SOFTWARE SUPPLIERS IN CANADA

COUNTRY OF OWNERSHIP	TYPE OF VENDOR	LOCATION											
		Ontario					Quebec	B.C.		Alberta		NFLD.	
		Metro Toronto	Metro Ottawa	Kichener	London	Barrie	Metro Montreal	Van.	Vic.	Edm.	Cal.	St. John's	TOTAL
United States	Mainframe	6											6
	Minicomputer	8	1										9
	Microcomputer	1				1							2
	Word Processor	2											2
	Serv. Bureau	2								1			3
	OEM, System-house, Dis. & 3rd Party Spec.												0
	Ter & Per Spec.	1											1
	Software Houses	3											3
	Sub-total	23	1			1				1			26
Canada	Mainframe												0
	Minicomputer	1											1
	Microcomputer		1										1
	Word Processor						1						1
	Serv. Burea	3			1		1		1		1	1	8
	OEM, System-house, Dis. & 3rd Party Spec.	1	1				3						5
	Ter& Per Spec			1									1
	Software House	1	3				1	2	1				8
	Sub-total	6	5	1	1		6	2	2		1	1	25
Netherlands	Word Processor						1						1
	Sub-total						1						1
United Kingdom	Minicomputer	1											1
	Sub-total	1											1
TOTAL		30	6	1	1	1	7	2	2	1	1	1	53

EXHIBIT 6-5

Software Suppliers in the Canadian Software Industry by Country of Ownership

Country	Top EDP Companies* Reporting Software Revenues	Key 53 Software Suppliers
United States	40	26
Canada	52	25
United Kingdom	2	1
Bahamas	1	-
France	1	-
Italy	1	-
Netherlands	1	1
Total	98	53

Source: Evans Research Corporation

* As provided in Evans Research Corporations' EDP In-Depth Report entitled, "The Top Computer Companies in Canada."

Of the total 1981 software revenues of $608.27 million, 66.5% ($404.3 million) resulted from sales by foreign owned firms. Twenty-five Canadian owned software firms amongst these 53 top suppliers had revenues amounting to $139.9 million in 1981 (23% of the 1981 total software market). In the second category, the 463 software firms with revenues of less than $1 million, over 79% of these (366 firms) are Canadian owned and the majority of the remainder are US owned. Canada's leading 1982 software and EDP consulting firms are shown in Exhibit 6-6.

6.1.2 Software Employment

It is estimated that there are more than 4,000 persons employed in Canada's software industry, including persons directly involved in developing, maintaining or modifying software systems, but not including clerical, managerial, marketing and sales staff.

The top 53 software suppliers employed 2,566 people in software development (or 60.3% of the total number of employees involved in developing software in 1982).

The other 463 cottage type/small business software companies employ an average of three persons each in software development, and as shown in Exhibit 6-7, the biggest employers of software development are Canadian service bureaus which employ a total of 886 persons (20.8% of the total number of employees in software development in 1982).

EXHIBIT 6-6

Canada's Leading Software and EDP Consulting Companies*

Company Name	% of Total 1982	% of Total 1981	EDP Revenues** 1982	EDP Revenues** 1981	% Growth 82/81
Systemhouse Ltd.	28.2	28.9	39.8	30.5	30
DMR & Associates	23.4	25.8	33.1	27.2	22
Quasar Systems Ltd.	13.2	10.5	18.6	11.1	68
Sydney Development Corp.	5.7	4.6	8.0	4.9	63
Bailey & Rose	5.2	5.1	7.3	5.4	35
MSA Canada Inc.	3.6	2.9	5.1	3.1	65
Cincom Systems of Canada	3.5	3.2	5.0	3.4	47
Computech Consulting Canada	3.4	3.7	4.8	3.9	23
Le Groupe BST Ltd.	2.3	3.3	3.2	3.5	(9)
LGS Data Processing Cons.	2.0	1.7	2.8	1.8	56
ISS Info. System Services	2.0	3.5	2.8	3.7	(24)
Pansophic Systems of Canada	1.7	1.9	2.4	2.0	20
University Comp. Co. Can.	1.6	1.8	2.2	1.9	16
Applied Data Research Can.	1.5	1.0	2.1	1.1	91
Dyad Computer Systems Inc.	1.5	2.0	2.1	2.1	0
Polaris Technology Corp.	1.5	-	2.1	-	-
Total			141.4	105.6	34

* Includes those firms who derive the bulk of their revenues from EDP related consulting and/or sale/licensing of custom and/or "packaged" software.

** Figures are $ Canadian millions.

Source: EDP In-Depth Reports, June 1983.

EXHIBIT 6-7

1982 Estimated Canadian Employment in Software Development in the Top 516 Suppliers

Type of Software Vendor	Total No. of Employees	No. of Employees in Software Development	% in Software Development
Mainframe	17,765	675	3.8
Minicomputer	7,310	731	10.0
Microcomputer	2,450	69	2.8
Word Processing	3,775	189	5.0
OEMs, Systemhouses, Distributors and 3rd Party Companies	1,850	93	5.0
Service Bureaus	8,604	886	10.3
Terminal and Peripheral Equipment Specialists	1,375	69	5.0
Software Houses and EDP Consulting Firms	1,990	343	17.3
Other*	1,600	1,200	75.0
Total	38,201	4,255	11.1

* Estimated 463 firms with total software revenues less than $1 million.

Source: Evans Research Corporation

6.1.3 Market Size and Growth

Although Evans has found a 33% annual growth in the Canadian software market size from 1980 to 1981, this figure includes only sales of "unbundled" software and excludes "bundled" and special purpose software, such as that for telecommunications and video games.

Evans predicts that the software market share of the foreign hardware manufacturers will decrease from 56% in 1980 to 38% in 1990 because of increased competition from software houses, EDP consultant firms and the traditional tendency of the hardware manufacturers to concentrate more on systems software rather than applications. (However, as we have seen, many of the hardware manufacturers are simply vertically integrating by buying out the independent software firms).

Evans also predicts that the share of the total Canadian software market accruing to software houses and EDP consulting firms will grow from 30% of the total 1980 market to more than 42% by 1990.

Canada's service bureaus comprised only 11% of the total 1980 software market and 10.5% of the 1981 market, but Evans predicts that this group's share of the market will increase to 16% by 1990 because the general decline of service bureaus will force them to become more involved in software production, and they already have the necessary development expertise and firm infrastructure to enter this field.

Although the study excludes telecommunications software from its data base, it predicts that in Canada there will be a constant growth in systems software such as network management/communications software, control software, installation management software, and data and terminal access systems.

In summary, the largest US software firms developing packages have sales in the range of approximately $100 million annually, but over 90% of US software firms in the package business have revenues of less than $3 million yearly. The Canadian software industry, in contrast, has only three firms in the range of $20-$40 million annual sales, and over 90% of the companies have sales below $5 million annually.

One study has recently summarized that the Canadian industry is:

> "... suffering all of the strains of fast growth business, including shortage of professional management, ill-defined market and project strategies, and a highly fragmented industry structure. For the Canadian companies to be successful they must take an export market orientation, an approach that is costly for small companies." (85)

Market forecasts to 1990 for the Canadian software industry, by software and vendor type are presented in Exhibits 6-8 and 6-9.[86]

Evans asked 30 firms to identify what they consider to be areas of opportunity in software development. Half of the respondents identified microcomputer software as the largest opportunity area, with data base management, word processing, home financing spread sheets, decision support, productivity tools and communications software being mentioned.

However, more than half (17 respondents) insisted the following were major opportunity areas: office automation, computer learning courseware, CAD/CAM, telecommunications software, factory automation and Telidon.

Seventeen respondents also indicated the following specific vertical markets provided the best software opportunities: resources, agriculture, health care, manufacturing, communications and banking.

EXHIBIT 6-8

CANADIAN SOFTWARE INDUSTRY FORECAST TO 1990

SOFTWARE TYPE	1980	1981	1982	1983	1984	1985
SYSTEM SOFTWARE						
- $ Millions	128	174	233	313	425	568
- % Growth		36	34	35	36	34
- Percentage of Total Market	28	29	29	30	30	31
APPLICATIONS SOFTWARE						
- $ Millions	114	161	223	310	434	596
- % Growth		41	38	39	40	37
- Percentage of Total Market	25	27	28	30	31	33
SYSTEM DEVELOPMENT						
- $ Millions	215	273	341	428	540	669
- % Growth		27	25	26	26	24
- Percentage of Total Market	47	45	43	41	39	37
TOTAL	457	608	797	1,052	1,399	1,833
- % Growth		33	31	32	33	31

SOFTWARE TYPE	1986	1987	1988	1989	1990	Average Annual Growth Rate 1980-1990
SYSTEM SOFTWARE						
- $ Millions	741	952	1,202	1,493	1,824	
- % Growth	30	28	26	24	22	31%
- Percentage of Total Market	32	32	33	33	34	
APPLICATIONS SOFTWARE						
- $ Millions	798	1,049	1,356	1,721	2,146	
- % Growth	34	32	29	27	25	34%
- Percentage of Total Market	34	36	37	39	40	
SYSTEM DEVELOPMENT						
- $ Millions	807	955	1,107	1,256	1,395	
- % Growth	21	18	16	14	11	21%
- Percentage of Total Market	34	32	30	28	26	
TOTAL	2,346	2,956	3,665	4,471	5,365	
- % Growth	28	26	24	22	20	28%

Source: Evans Research Corporation

EXHIBIT 6-9

ESTIMATED SOFTWARE REVENUES BY VENDOR TYPE IN THE CANADIAN SOFTWARE MARKET
FORECAST TO 1990

Vendor Type	1986	1987	1988	1989	1990	Average Annual Growth 1980-1990
HARDWARE MANUFACTURERS						
-$ Millions	1,056	1,275	1,512	1,761	2,012	
-% Growth	23	21	19	16	14	23%
-Percentage of Total Market	45	43	41	39	38	
SERVICE BUREAUX						
-$ Millions	318	419	542	688	858	
-% Growth	34	32	29	27	25	33%
-Percentage of Total Market	14	14	15	15	16	
SOFTWARE HOUSES AND EDP CONSULTING FIRMS						
-$Millions	879	1,141	1,457	1,827	2,253	
-% Growth	32	30	28	25	23	33%
-Percentage of Total Market	37	39	40	41	42	
OEM'S,SYSTEMHOUSES, DISTRIBUTORS AND THIRD PARTY COMPANIES						
-$Millions	62	80	103	130	161	
-% Growth	33	31	28	26	24	33%
-Percentage of Total Market	3	3	3	3	3	
TERMINAL AND PERIPHERAL EQUIPMENT SPECIALISTS						
-$Millions	35	44	55	67	80	
-% Growth	36	26	24	22	20	34%
-Percentage of Total Market	2	2	2	2	2	
TOTAL						
-$Millions	2,346	2,956	3,665	4,471	5,365	
-% Growth	28	26	24	22	20	28%

Source: Evans Research Corporation

EXHIBIT 6-9
(continued)

ESTIMATED SOFTWARE REVENUES BY VENDOR TYPE IN THE CANADIAN SOFTWARE MARKET
FORECAST TO 1990

	Actual			Forecast		
Vendor Type	1980	1981	1982	1983	1984	1985
HARDWARE MANUFACTURERS						
-$ Millions	258	331	419	533	683	860
-% Growth		28	26	27	28	26
-Percentage of Total Market	56	54	53	51	49	47
SERVICE BUREAUX						
-$ Millions	51	64	89	124	173	238
-% Growth		27	39	39	40	37
-Percentage of Total Market	11	11	11	12	12	13
SOFTWARE HOUSES AND EDP CONSULTING FIRMS						
-$Millions	135	194	263	359	493	666
-% Growth		44	36	37	37	35
-Percentage of Total Market	30	32	33	34	35	36
OEM'S, SYSTEMHOUSES, DISTRIBUTORS AND THIRD PARTY COMPANIES						
-$Millions	9	13	18	25	34	46
-% Growth		43	37	38	38	36
-Percentage of Total Market	2	2	2	2	2	3
TERMINAL AND PERIPHERAL EQUIPMENT SPECIALISTS						
-$Millions	4	6	9	13	18	26
-% Growth		42	42	42	43	40
-Percentage of Total Market	1	1	1	1	1	1
TOTAL						
-$Millions	457	608	797	1,052	1,399	1,833
-% Growth		33	31	32	33	31

Source: Evans Research Corporation

6.1.4 Small Software Firms

Evans has concentrated on Canada's 53 top software suppliers. But what of the several hundred small firms with less than $1 million annual revenue? These types of firms were the subject of a 1984 study of BC's software industry[87] which, with the exception of Ontario, is perhaps characteristic of local software industries across Canada.

Most of Canada's software writers in fact are located in Ontario (in Ottawa's Silicon Valley and in Metro Toronto). Exhibit 6-10 presents the provincial percentages of total Canadian software purchases in 1981. Ontario plus Quebec together accounted for over 77% of all software purchases, followed by BC and Alberta.

In the BC study, of 74 respondent firms, 37 were involved in pure software development, eight in the development of computer hardware as a main line of business, 12 in the production of turnkey systems, nine service bureaus and eight "other types" of firms.

With respect to employment levels, 54% of respondent firms were small (had five or less employees) while 30% had six to fifteen employees, and 16% had more than fifteen employees.

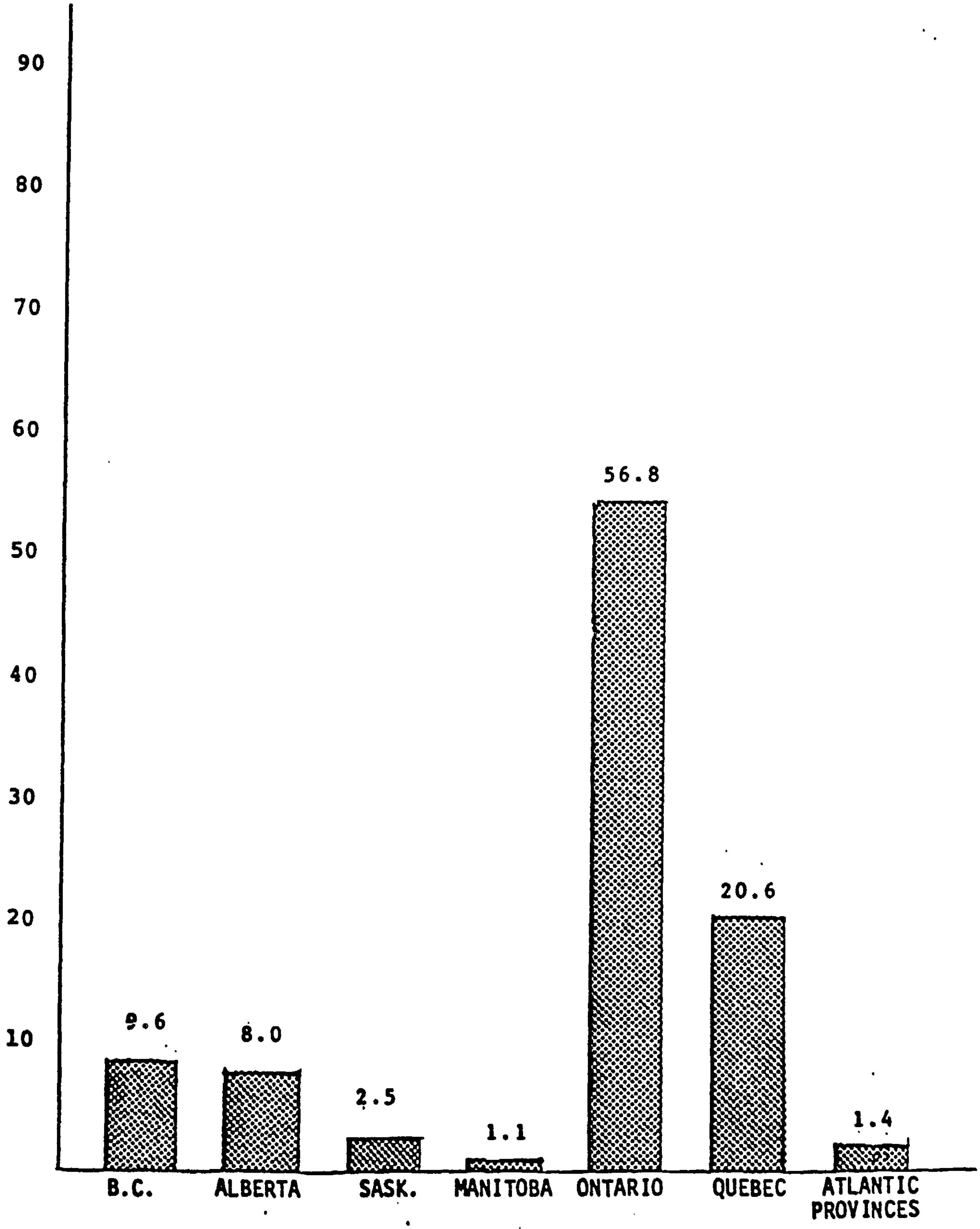
These "large" firms provided 61% of the total software employment of BC in 1983, while medium sized firms provided 26% of total software employment.

Anticipated employee growth of responding small firms varied from an average of 2.5 employees in 1983 to an average of 6.5 by 1986, while the equivalent figures for medium sized and large sized companies were 9.4 to 14.1 employees and 39.5 to 79.4 employees, respectively.

In terms of actual employee growth (Exhibit 6-11), large firms are growing at faster rates than smaller ones.

EXHIBIT 6-10

Provincial Percentages of Total Canadian Software Purchases, 1981



Source: Statistics Canada 63-222-1981

EXHIBIT 6-11

Employee Growth Rate

% Increase 1980-83	% Small Co.'s (0-5)	Medium Co.'s (6-15)	Large Co.'s (15+)	Total
Increased over 50%	25.0	32.0	50.0	31.1
Increased 10% to 49%	7.5	27.2	33.3	17.6
No significant change	30.0	18.2	-	21.6
Decreased 10% to 50%	15.0	9.1	8.3	12.2
Decreased over 50%	2.5	13.5	-	5.4
No response	20.0	-	8.4	12.2

Source: Touche Ross & Partners

6.1.4.1 Software Revenues

In terms of revenues (Exhibit 6-12), over 63% of respondent firms had revenues of less than $100,000 and 13 respondent firms had revenues of less than $50,000 in 1982.

The authors of this study speculate that most BC software suppliers are simply too small to be able to fund the current development and marketing necessary to reap market opportunities.

Exhibit 6-13 compares revenue percentages by type of software between BC, the overall Canadian industry and the US industry. Of particular significance is the fact (not shown on the graph) that the large BC software firms are highly dependent on production of custom software, with about 64% of their revenues derived from custom software compared to 48% for the entire BC industry and 45% for Canada as a whole.

Given the trend toward application packages, it is significant that the large firms did not report strong intentions to move away from custom to package development.

Exhibit 6-14 presents revenue percentage by hardware and firm size. Small BC software producers are clearly dependent on minicomputers.

Also, it is mainly smaller firms which are writing software for microcomputers, and these firms usually lack the financial/marketing resources for commercial development. In fact, only a small percentage of the large software firms in BC have decided to produce software predominately for microcomputers.

EXHIBIT 6-12

Revenue Levels of BC Software Suppliers

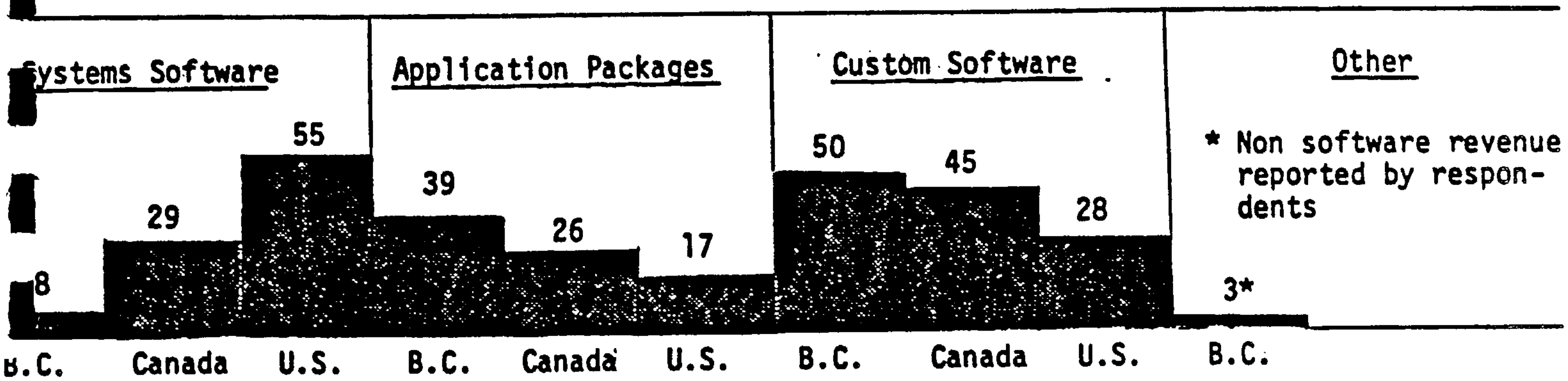
Companies With BC Operations Only	No. of Companies	% of Respondents at Various Levels of BC Software Revenues
1982 Revenue Category:		
Very small (under $100,000)	26	63%
Small ($100,000 - 499,999)	9	22%
Medium ($500,000 - 999,999)	6	15%
Large (over $1,000,000)	0	-
No response	0	-
BC-Based Companies With Other Branches:		
Very small	0	-
Small	7	58%
Medium	2	17%
Large	3	25%
Ex-BC-Based Companies With Local Representation:		
Very small	3	25%
Small	3	25%
Medium	3	25%
Large	3	25%

Source: Touche Ross & Partners

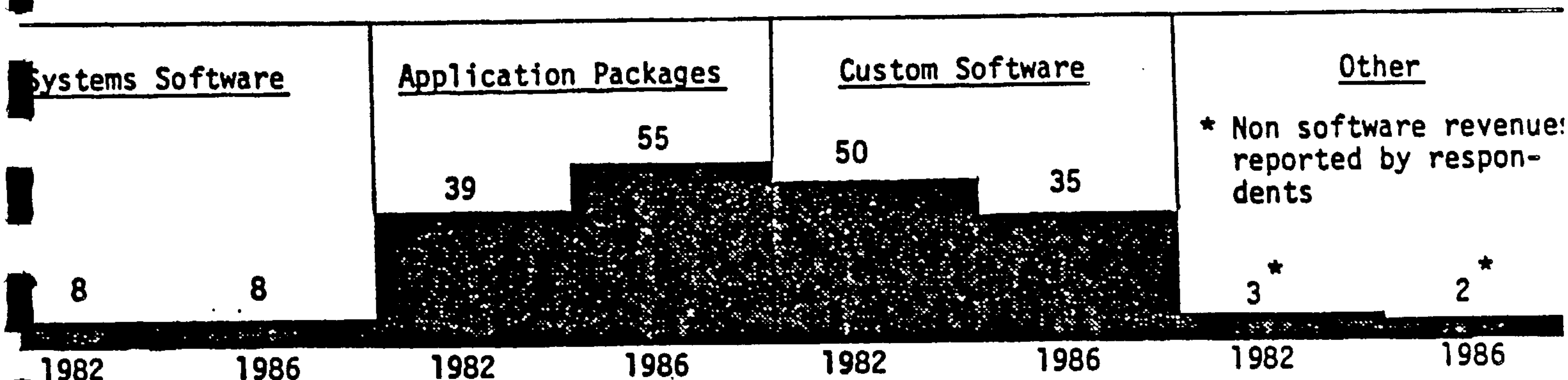
EXHIBIT 6-13

Percentages of Software Revenues by Software Type

TOTAL B.C. INDUSTRY VS. TOTAL CANADA AND U.S. MARKETS



% OF REVENUE DISTRIBUTION
TOTAL B.C. INDUSTRY 1982 VS 1986



Sources: B.C. - Touche Ross & Partners Industry Survey
Canada - Evans Research
U.S. - International Data Corporation

EXHIBIT 6-14

% of 1982 Revenues

Hardware Size	V Small Co.'s	Small Co.'s	Medium Co.'s	Large Co.'s	Total* Canada
Mainframe	19	15	35	39	73
Mini	45	57	44	51	22
Micro	36	28	21	10	5
Total	100	100	100	100	100

Source: Touche Ross & Partners

Exhibit 6-14A presents the distribution of 1982 software revenue per geographic source. The study notes that it is not obvious from percentages of this exhibit that total revenues derived from outside Canada by the six largest firms are several times the amount of outside revenues of all other firms combined. In other words, firms with revenues of less than $1 million have virtually no exports to the US or to the international markets. Furthermore, questions asked about intentions to move into new markets brought only "moderately planned shifts".

Advertising and marketing of the BC software producers is almost non-existent, relying predominately on personal contact, referrals and repeat business. Marketing has been tailored to local industry, directed predominately toward custom products.

6.1.4.2 Development Efforts

Within the larger firms in the BC sample, costs of developing new packages and costs for the modification of existing packages consumed about 50% of typical annual resources. Packaged producers in BC are concentrating on vertical markets and on horizontal markets among traditional business software, such as payroll, accounting, etc. The key problems identified seem to be in the areas of marketing orientation, marketing know-how, marketing channels and financial resources.

6.1.4.3 Sources of Funding

Exhibit 6-15 presents sources of funding used by BC survey respondents. Most of these software producers relied on borrowing from banks or partners for financing. Over 80% of the respondents felt constrained by the lack of available financing, but none of the large firms reported financing problems. The report speculated that few respondents realized the magnitude of future marketing costs involved in packaged software production, and that there will be a real need for expanded funding from a variety of sources.

EXHIBIT 6-14A

Geographic Sources of BC Software Company Revenues

	Total Sample Average % of Revenue	Large Company Only % of Revenue
BC	81.3	75.0
Prairies, NWT, Yukon	4.0	4.3
Eastern Canada	7.3	.5
Washington, Oregon, California, Alaska	2.0	6.5
Rest of USA	4.5	5.7
Outside North America	.9	8.0

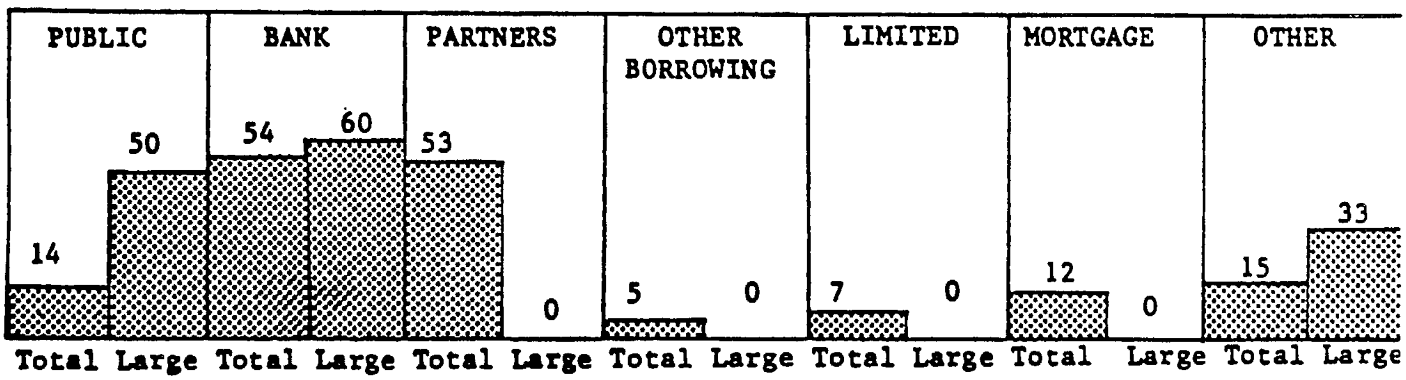
Source: Touche Ross & Partners

EXHIBIT 6-15

Current Sources of Funding Used by Survey Respondents

	Number of Respondents			
	TOTAL		LARGE CO'S ONLY	
	Number	Percentage	Number	Percentage
Public Stock	10	14%	3	50%
Bank Borrowing	40	54%	4	60%
Borrowing from Proprietor or Partners	39	53%	0	0
Other Borrowing	4	5%	0	0
Limited Partnership	5	7%	0	0
Chattel or other mortgaging	9	12%	0	0
Other	11	15%	2	33%

EXPRESSED IN PERCENTAGE



6.1.4.4 Summary

The report is not optimistic about development possibilities in BC's software industry. The list of perceived advantages is remarkably short, being limited to attractiveness of living environment, proximity to BC's forest products' base, and to several large engineering companies operating on a world scale.

The perceived disadvantages emerging from this study involved a weak competitive position (few firms being in a financial position to make the magnitude of investments necessary to support software products to the marketplace); the considerable distance from the major markets (software development firms must have, in spite of the rhetoric about operating at a distance, close customer interaction); the distance from resources for developing products for major vertical markets (all places where software centres are developing are near major industries with which software firms can work developing and testing new products). Since BC has access to only a few of these industries, such as engineering, forestry and education, and BC's resource industries are going through a severe recession, the range of key successes in vertical markets, the study predicts, will be limited until new industries locate in BC or the BC software firms are large enough to set up branch offices in the major centres near these industries. (The same observation can be made of Alberta's software firms, whose major driving industry -- petroleum resources -- has been decimated by the National Energy Policy.) Other disadvantages include a comparatively low concern for expansion, inadequate marketing expertise and orientation, and a low level of access to government business. British Columbia Systems Corporation until now has been the main supplier of software for provincial government departments, but most software developers have experienced comparatively little benefit of provincial government procurement. In addition, a small percentage of BC software suppliers have retained "significant revenues" as a result of federal govenment procurement practices. Finally, a major, severe and constant constraint is the availability of financial resources.

In summary, the report notes that the "typical" BC software supplier:

- Is managed by a proprietor or a small group of partners whose backgrounds and interests are primarily technical;
- Lacks the marketing knowledge and expertise required to compete strongly in the packaged-oriented market of the eighties;
- Lacks the manpower resources required to carry out effective marketing programs;
- Does not have sufficient finances or borrowing power to invest heavily either in existing product development or in marketing.

Major directions of change suggested by the study involve increased marketing orientation and confidence, increased investment in development and marketing of software products with unique appeals to potential customers, and a greater emphasis on strategic business planning and management. However, the study remarks that individual changes within software companies along these lines will not be enough and there is a real need for a broader scale cooperative program between industry and government bodies.

6.1.5 The Canadian Computer Industry

Evans and others have predicted that in 1984 the Canadian computer industry will grow by 19%, well below the 25% annual growth rate of several years ago. Personal computers will comprise almost $8.13 billion of the total predicted domestic computer market, and IBM continues to dominate the Canadian market, receiving 25¢ for each dollar spent for the industry's hardware and software. IBM Canada reported 1982 revenues of $2.46 billion, compared to 1981 revenues of $2.21 billion. Comparative profits in these two years were $248 million and $172 million, respectively.

Given IBM's current drive with the IBM PC, it seems likely that very shortly IBM and it's compatibles will control much of the microcomputer market. This dominance will also increase the predominance of software vendors which furnish software which runs under the IBM PC's "DOS" operating system. Almost all business software now runs under DOS or it's clone, the MS-DOS. The range and versatility of software written for this operating system is increasing rapidly and it has many features which resemble those of AT&T's Unix operating system, which is used for minicomputers and mainframes. Also in March 1984, AT&T introduced six (personal and mini) computers, all based on UNIX. Paradoxically, Nigel Kelly has observed, vendors which supply IBM-compatible microcomputers have supported IBM's rapid rise to dominance:

> "By offering superiority in the minor product features but also compatability in the major ones, these vendors leave IBM in control of the strategic direction." (87)

Exhibit 6-16 presents Canada's top 100 computer firms. As can be seen, most of the top firms are subsidiaries of US based multinationals. Most of the larger Canadian controlled firms are computer service bureaus.

The Canadian microcomputer industry will grow from $530 in 1983 to $2.36 billion[88] in 1988. Exhibit 6-17 presents a breakdown of the Canadian microcomputer market by systems units, peripherals, add-ons, applications and systems software. Evans notes that perscom software is expected to grow at average rates of 39% annually during the next five years, and in late 1988 sales of systems and applications software for perscoms will be almost as large as the entire microcomputer industry in 1982.

6.1.6 Federal Government Programs to Support Software

Although there are no Canadian federal government programs directed specifically to software firms, there are a number of programs with "high technology components" which may or may be applicable to software firms:

- The Office Communications Systems Program, initiated by the Department of Communications, is attempting to develop Canadian office automation capabilities in industry and encourage R&D in this area. Thus far, however, activities of this program have been mainly limited to field trials in selected government departments and in spite of rhetoric to the contrary, the emphasis thus far has been on hardware rather than software.

EXHIBIT 6-16

The Top 100 Companies in the Canadian Computer Industry

	Company Name	Fiscal Year End	Ownership	Total Revenues 1982	EDP Revenues *1 1982	1981	1980
1.	IBM Canada Ltd.	Dec. 31	U.S.	2,204.0	1,890.0	1,517.7	1,216.5
2.	Digital Equipment of Canada Ltd.	July 1	U.S.	294.9	294.9	252.6	163.7
3.	Control Data Canada Ltd.	Nov. 30	U.S.	231.0	231.0	181.3	162.6
4.	Philips Information Systems Ltd.	Dec. 31	NL	199.6	199.6	140.0	100.3
5.	AES Data Inc.	Dec. 31	Can.	188.1	188.1	172.8	155.4
6.	NCR Canada Ltd.	Nov. 30	U.S.	180.0	162.0	152.4	150.1
7.	Canada Systems Group Ltd. *2	Dec. 31	Can.	140.6	140.6	136.7	116.3
8.	Sperry Inc.	Mar. 31/83	U.S.	354.0	140.4	141.0	132.0
9.	Honeywell Ltd. (Information Systems)	Dec. 31	U.S.	340.0	110.0	94.4	85.0
10.	Burroughs Canada *3	Nov. 30	U.S.	121.0	108.5	114.0	94.0
11.	Hewlett-Packard (Canada) Ltd.	Oct. 31	U.S.	182.0	101.9	81.0	63.5
12.	Datacrown Inc.	Dec. 31	Can.	88.0	88.0	86.2	68.6
13.	STC Canada Inc.	Dec. 31	U.S.	84.7	84.7	56.4	28.8
14.	Amdahl Ltd.	Dec. 31	U.S.	77.0	77.0	72.6	43.0
15.	Wang Canada Ltd.	June 30	U.S.	64.7	64.7	44.5	21.0
16.	B.C. Systems Corp.	Mar. 31/83	Can.	63.0	63.0	57.9	49.1
17.	Radio Shack (Tandy Electronics Ltd.)	June 30	U.S.	197.7	61.3	39.7	22.3
18.	Memorex Canada *3	Nov. 30	U.S.	57.4	57.4	40.5	30.8
19.	Gandalf Technologies Inc.	July 31	Can.	53.3	53.3	40.2	26.1
20.	MAI Canada Ltd.	Sep. 30	U.S.	52.0	52.0	47.1	39.4
21.	I.P. Sharp Associates Ltd.	Dec. 31	Can.	51.6	51.6	42.0	35.3
22.	Xerox Canada Inc.	Dec. 31	U.S.	596.0	47.7	40.2	30.9
23.	Geac Computer Corporation Ltd.	Apr. 30/83	Can.	47.0	47.0	35.6	23.6
24.	Systemhouse Ltd.	Aug. 31	Can.	39.8	39.8	30.5	19.3
25.	CGE Co. Ltd. (Information Services)	Dec. 31	U.S.	1,633.8	37.0	31.0	26.0
26.	Nabu Manufacturing Corporation *4	July 3	Can.	35.0	35.0	29.9	18.2
27.	DMR & Associates	May 31/83	Can.	33.1	33.1	27.2	18.7
28.	Data General (Canada) Inc.	Sep. 25	U.S.	32.4	32.4	29.0	23.0
29.	ESE Limited	Dec. 31	U.S.	30.4	30.4	10.0	5.4
30.	Commodore Business Machines Ltd.	June 30	Bah.	n/a	30.0	18.0	7.4
31.	Northern Telecom Ltd.	Dec. 31	Can.	3,035.5	29.0	26.2	21.0
32.	L'Industrielle-Services Techniques Inc.	Nov. 30	Can.	28.4	28.4	23.8	20.4
33.	Mohawk Data Sciences Canada Ltd.	Apr. 30/83	U.S.	26.8	26.8	22.3	16.5
34.	Electrohome Ltd.	Dec. 31	Can.	196.3	25.0	20.0	10.0
35.	Apple Canada Inc.	Sep. 24	U.S.	24.2	24.2	20.1	0.0
36.	Datapoint Canada Inc.	June 30	U.S.	23.5	23.5	n/a	n/a
37.	ADP Automatic Data Processing Inc.	June 25	U.S.	23.0	23.0	20.0	16.7
38.	Cherney Hills Inc. (CHI Company)	Dec. 31	U.S.	22.4	22.4	15.9	20.7
39.	SaskCOMP	Dec. 31	Can.	21.7	21.7	16.7	12.8
40.	Lanpar Technologies Inc.	Jan. 31/83	Can.	21.4	21.4	16.4	15.0
41.	Altel Data	Dec. 31	Can.	n/a	21.0	15.0	12.0
42.	Tandem Computers Canada Ltd.	Sep. 30	U.S.	20.5	20.5	15.7	5.2
43.	Quasar Systems Ltd.	Aug. 31	Can.	18.6	18.6	11.1	5.7
44.	National Advanced Systems *5	Mar. 31/83	U.S.	18.0	18.0	15.0	12.0
45.	General DataComm Ltd.	Sep. 30	U.S.	17.2	17.2	13.9	12.5
46.	Olivetti Canada Ltd.	Dec. 31	I	64.0	17.0	15.5	11.0
47.	Intergraph Systems Ltd.	Dec. 31	Can.	16.6	16.6	16.1	12.4
48.	Ahearn & Soper Inc.	Dec. 31	Can.	16.0	16.0	14.0	12.8
49.	Prime Computer of Canada Ltd.	Dec. 31	U.S.	16.0	16.0	13.0	10.0
50.	Manitoba Data Services	Mar. 31/83	Can.	15.7	15.7	15.5	13.0
51.	Comterm Inc.	Jan. 31/83	Can.	15.3	15.3	9.5	6.8
52.	ICL Computers Canada Ltd.	Sep. 30	U.K.	14.0	14.0	15.0	15.3
53.	Tulsa Computer Products Ltd. (Telex)	Mar. 31/83	U.S.	13.9	13.9	9.0	6.0
54.	Four Phase Systems Ltd.	Dec. 31	U.S.	13.1	13.1	13.4	12.4
55.	ITT Courier (ITT Canada Ltd.)	Dec. 31	U.S.	12.7	12.7	9.5	7.2
56.	Greyhound Computer of Canada Ltd.	Dec. 31	U.S.	12.3	12.3	5.9	5.1
57.	Computer Sciences Canada Ltd.	Mar. 31/83	U.S.	12.0	12.8	12.2	11.2
58.	Matrox Electronic Systems Ltd.	Mar. 31/83	Can.	11.0	11.0	8.7	4.7
59.	BTS Systems Ltd.	Apr. 30/83	Can.	11.0	11.8	9.0	6.8
60.	NCR Comten Inc.	Dec. 31	U.S.	10.4	10.4	5.2	2.5

EXHIBIT 6-16
(continued)

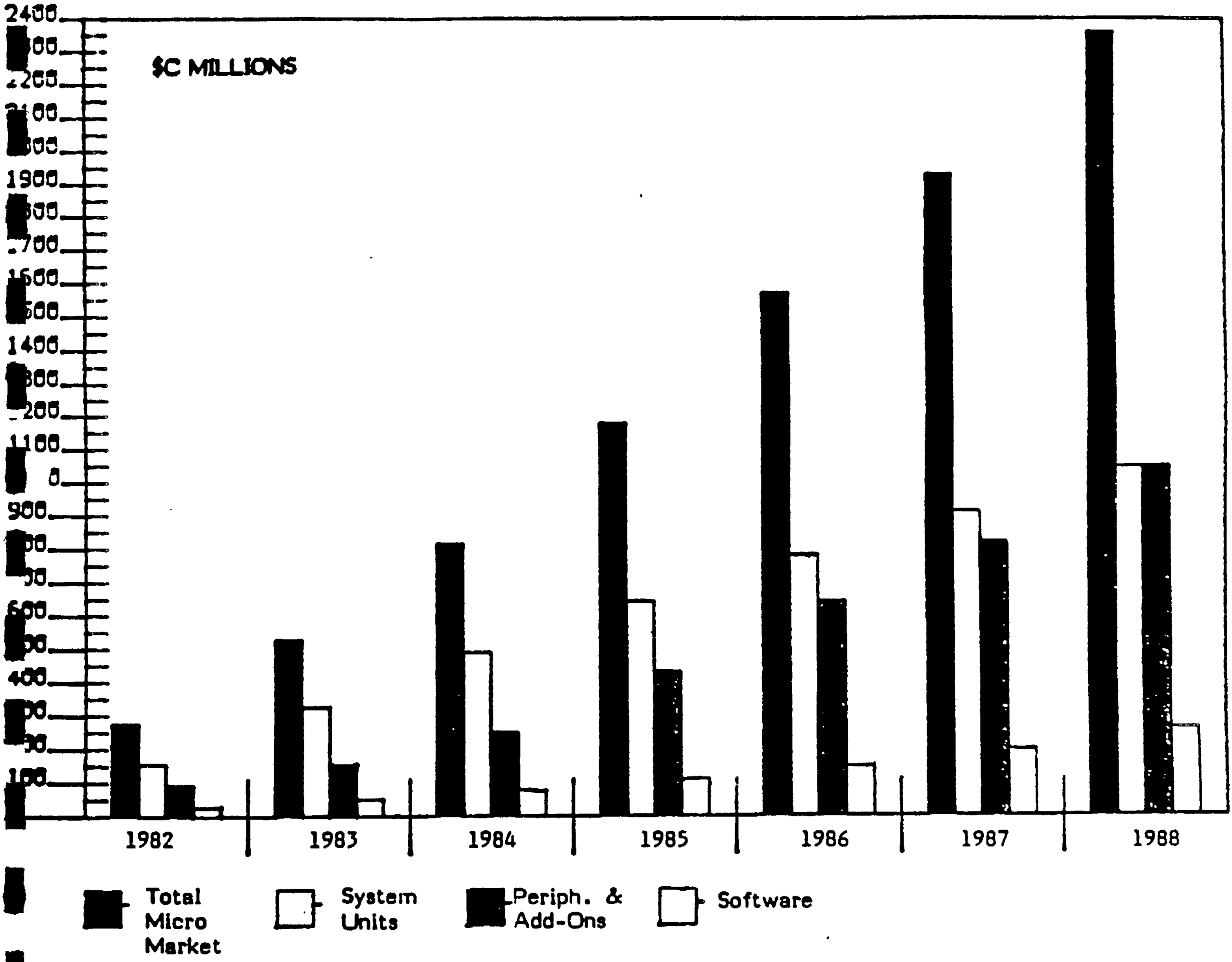
61.	Comshare Limited	Dec. 31	Can.	10.3	10.3	10.2	7.6
62.	Dataline Inc.	Dec. 31	Can.	10.3	10.3	9.9	8.2
63.	Nelma Data Corporation	June 30	Can.	10.1	10.1	7.6	4.2
64.	Data Terminal Mart	June 30	Can.	10.0	10.0	8.5	7.5
65.	Norpak Corporation	June 30	Can.	10.0	10.0	7.2	5.0
66.	Texas Instruments Canada Ltd.	Dec. 31	U.S.	n/a	10.0	10.0	10.8
67.	Digitech Ltd.	June 30	Can.	51.0	9.7	9.0	7.9
68.	Develcon Electronics Ltd,	Aug. 31	Can.	9.7	9.6	6.7	3.1
69.	Reynolds & Reynolds	Sep. 30	U.S.	15.4	9.5	11.7	12.6
70.	Hamilton Rentals (Hamilton Group Ltd.)	Apr. 30/83	Can.	9.4	9.4	7.0	3.8
71.	Centronics Canada Inc. *6	Dec. 31	U.S.	9.2	9.2	9.1	6.8
72.	Newfoundland & Labrador Comp. Serv. Ltd.	Mar. 31/83	Can.	9.2	9.2	8.4	7.3
73.	Real Time Datapro Ltd.	Feb. 28/83	Can.	9.5	9.0	6.4	5.1
74.	Datamex Ltd.	June 30	Can.	8.3	8.3	5.7	4.8
75.	Thorne Riddell	Dec. 31	Can.	145.0	8.1	6.1	4.0
76.	Datatech Systems Ltd.	Aug. 31	Can.	9.1	8.0	6.5	4.8
77.	Perkin-Elmer Canada Ltd. (Data Systems)	June 30	U.S.	n/a	8.0	6.8	6.1
78.	Recognition Equipment (Canada) Ltd.	Oct. 31	U.S.	8.0	8.0	8.1	6.3
79.	Sydney Development Corporation	Mar. 31/83	Can.	8.0	8.0	4.9	0.6
80.	Zentronics (Westburne Industries)	Mar. 31/83	Can.	32.0	8.0	8.0	4.0
81.	ACT Computer Services Ltd.	Dec. 31	Can.	7.8	7.8	7.5	6.8
82.	Alphatext (Ronalds-Federated Ltd.)	Dec. 31	Can.	7.8	7.6	4.9	4.7
83.	CTS Computer Systems Inc. *7	Dec. 31	Can.	7.4	7.4	7.0	4.7
84.	Bailey & Rose	June 30	Can.	7.3	7.3	5.4	2.9
85.	Cableshare Inc.	Aug. 31	Can.	7.2	7.2	6.6	4.2
86.	Cybernex Ltd.	Jan. 31/83	Can.	7.1	7.1	5.2	3.1
87.	Riley's Datashare International Ltd.	May 31	Can.	7.8	7.0	5.8	5.1
	Tektronix Canada Inc.	May 28	U.S.	32.0	7.0	9.0	6.0
	TRW Canada Ltd. (TRW Data Systems) *8	Dec. 31	U.S.	n/a	7.0	28.0	22.0
9[illegible].	Computervision Canada Inc.	Dec. 31	U.S.	6.9	6.9	5.8	2.8
91.	Dasco Data Products	Sep. 30	Can.	6.9	6.9	6.2	5.1
92.	Hospital Computing Services of Ontario	Mar. 31/83	Can.	6.9	6.9	6.0	4.9
93.	National Datacentres Corp.	Mar. 31/83	Can.	6.0	6.0	6.3	5.2
94.	Comtecb Group International Ltd.	June 30	Can.	5.9	5.9	5.8	5.7
95.	Nixdorf Canada Ltd.	Dec. 31	FRG	5.5	5.5	3.5	4.0
96.	Comcheq Services Ltd. *9	May 31	Can.	5.4	5.4	4.8	3.6
97.	M.I.C.R. Systems Ltd. *10	Nov. 30	Can.	5.1	5.1	6.1	5.1
98.	MSA Canada Inc.	Dec. 31	U.S.	5.1	5.1	3.1	3.2
99.	Perle Systems Ltd.	May 31/83	Can.	5.1	5.1	3.9	1.7
100.	Cincom Systems of Canada Ltd.	Sep. 30	U.S.	5.0	5.0	3.4	2.2
	Total				5,412.0	4,477.3	3,516.7
	% Growth				21	27	

Notes:

*1 Figures in $ Canadian millions, for the fiscal year ending in year indicated. Domestic and export EDP-related are include
*2 CSG's reported revenues are: 1982 - $127.6 million (includes 8 mths. of Computel Systems Ltd.); 1981 - $95.4 million (restated); 1980 - $77.9 million. CSG survey revenues include: estimated revenues of Computel Systems Ltd. of $13.0 million for the 4 mths. ending April 1982; 1981 fiscal year revenues of $41.3 million and 1980 fiscal year revenues of $30.[illegible] million.
*3 Division of Burroughs Memorex Inc. of Canada. Memorex Canada changed its FYE from Dec.31 to Nov. 30
*4 Includes revenues for Consolidated Computer Inc. and Volker-Craig division
*5 Division of National Semiconductor
*6 Changed FYE from June 30 to Dec. 31
*7 Acquired by Mohawk Data Sciences Canada Ltd. in March 1983
*8 1980 and 1981 revenues include sales and service of Datapoint Corp. products, now distributed by Datapoint Canada Ltd.
*9 Calendar year revenues
*10 Changed FYE form Dec. 31 to Nov. 30

EXHIBIT 6-17

CANADIAN MICROCOMPUTER MARKET FOR SYSTEMS LESS THAN $15K



Source: Evans Research Corporation surveys and estimates.

- The Enterprise Development Program, under the auspices of the Department of Regional Industrial Expansion, promotes industrial growth of processing and manufacturing industries and concentrates on existing small firms which are involved in high risk projects.

- The Support for Technology Enhanced Productivity Program, administered by the Department of Regional Industrial Expansion, was designed to help small firms in rural areas to apply chip-based technologies to production processes.

- The Export Market Development Program, again administered by the Department of Regional Industrial Expansion, provides grants to Canadian exporters in the area of export marketing and will support export marketing costs of engineering services and software.

- The New Technology Employment Program will pay 75% (to a maximum of $290 weekly) for wages of scientific/technical personnel who are hired to do research and development. Administered by the National Research Council, the program focuses on furthering R&D and technological innovation.

- The Regional Development Incentives Program, administered by the Department of Regional Industrial Expansion, encourages the growth of small businesses through direct investment.

- The Industrial Research Assistance Program, administered by the Department of Regional Industrial Expansion and the National Research Council, provides firms with up to 50% of a total project cost, with the stipulation being that the firm has to conduct the project in connection with a Canadian university.

- The Industrial and Regional Development Program, also administered by DRIE, applying to both manufacturers and providers of services such as software, is structured so that incentives to firms are higher in "economically disadvantaged" areas. Direct financial support of between 25% to 75% of total project costs are given, depending on the type and location of project.

- Finally, the Small Business Loans Program, again administered by the Department of Regional Industrial Expansion, attempts to develop indigenous industries in Canada and provides financing to new businesses or significant technical expansions of existing firms which involve the purchase of new equipment.

Most of these programs relate only peripherally to software. To varying degrees, several programs have expanded their scope beyond manufacture to knowledge services -- engineering, consulting and software. Especially useful has been the PEMDI support for marketing costs. However, in general firms preferred the availability of tax/fiscal incentives and found these programs cost more to utilize than benefits. Tax/fiscal incentives are analyzed in chapter seven.

6.2 Brazil

The development policies of Brazil are generally considered to be a model for third world informatics policies. Larger than the United States, Brazil now is included in the top nine data processing markets in the world, and data processing sales exceeded $2.4 billion in 1983. The main manufacturers of Brazilian computers and 1980 estimates for data processing revenues are presented in Exhibits 6-18 and 6-19.

During most of the 60s there was a minimal amount of regulation of importation of computers, software and data processing products into Brazil, and it was only in the early 70s that the government began to regulate imports with the aim of creating a national software and computer industry.

EXHIBIT 6-18

Authorized Brazilian Manufacturers

Company	Model	Technology	Description
COBRA - Computadores e Sistemas	COBRA-300	COBRA	Up to 48KB cpu Small single-user micro Based on floppy disks
	COBRA-400	SYCOR (US)	Up to 64KB cpu minicomputer with a model oriented to data entry and MUMPS
	COBRA-330	COBRA	Up to 512KB cpu with future expansion to 1 MB
EDISA - Electronica Digital S.A.	ED-300	FUJITSU (Japan)	Up to 64KB cpu Similar to IBM System 3
LABO Electronics	LABO 8034	NIXDORF (West Germany)	Up to 256KB cpu Similar to Nixdorf 8870-1
SID - Sistemas de Informacao Distribuida S.A.	SID-5000	LOGABAX (France)	Up to 64KB cpu Similar to DEC PDP 11/34
SISCO - Sistemas e Computadores S.A.	SCC-5000	SISCO	Up to 64KB cpu Similar to DEC PDP 8
	MB-800	SISCO	Up to 256KB cpu Similar to DG NOVA 3

Source: Datamation, May 1981, pp. 192-7.

EXHIBIT 6-19

Major Brazilian DP Suppliers

1980 Ranking by Revenue	Company	Estimate DP Revenues (in $ million) 1979	1980	Capital Registered in Brazil ($ million)	No. of Employees (1000)
1	IBM	350	330	130	5
2	COBRA*	70	104	9	2
3	Burroughs	100**	100**	2**	2**
4	SID*	19	30	5	.8
5	LABO*	5	22	2	.4
6	EDISA*	7	13	3	.3
7	Sisco*	.6	10	3	.3
8	Globus*	.8	10	.8	1
9	Elebra Informatica*	.5	9	2	.1
10	Scopus*	5	9	.8	.4
11	Honeywell-Bull	7	8	3**	.1**
12	Polymax*	.4	7	.4	.2
13	Microlab*	2	6	1	.3
14	Coencisa*	5	5	2	.2
15	Prologica*	2	5	.05	.2

* Data on these companies furnished by DIGIBRAS, a federal-owned company that provides technical and financial support to national hardware manufacturers.

** Company-furnished data.

It was in 1974, in fact, that policy planners in Brazil began to formulate their national informatics policy, and within seven years of this date imported software has been banned from Brazil. Foreign joint ventures in Brazil had to set up Brazilian design and manufacturing facilities within five years of the initiation date of the joint venture, and the Brazilian domestic market for perscoms and minicomputers had been reserved for Brazilian manufacturers.

All of these policy goals, we shall see, were accomplished with basically no research and development or fiscal/tax incentives and exclusively involved national financing by Brazilian firms and the government.

Foreign firms manufacturing informatics products in Brazil were soon legally required to export a specific number of products for each one sold on the Brazilian market, and although there was considerable balking on the part of the multinationals concerning the decision of reserving the market for minicomputers and perscoms, the decision was soon accepted.

The first Brazilian computer policy agency which formulated and enacted these measures, CAPRE, was made a part of Brazil's National Security Council in 1979 and renamed The Special Secretariat for Informatics (SEI). All importation of computer equipment must now be approved by SEI, and there is presently an approximately two year waiting period for approval.

6.2.1 The Brazilian Computer Market

The Brazilian mainframe computer market is dominated by IBM and Burroughs. Given the two year delay approved by SEI, the growth of mainframe computers over the past three years has been on average around 15% per annum. Minicomputer growth, however, has averaged more than 200% per annum and by 1981 several Brazilian manufacturers began to produce perscoms. However, there are few manufacturers in Brazil of computerized industrial process control equipment, and the main inhibiting factor here is Portuguese software for industrial process control.

In spite of all of these efforts, Brazil still does not have a viable semi-conductor industry, but the present development strategy appears to focus on creating a national market for computer terminals which in turn will (supposedly) generate demands for components on the part of Brazilian manufacturers, since this market is protected by tariffs. However, as Juan Rada has remarked:

> "The cost of such a strategy rules out competition in the international market and means that a policy of technical obsolescence closely related to the function of the equipment has to be adopted rather than a policy of keeping pace with the speed of innovation. In addition, it applies to slow diffusion of computer technology due to the higher price/performance of the equipment." (89)

Brazilian policy measures to enhance national software development, subsequently examined, include:

- commercialization and promotion of nationally developed software
- development of courses and training on software in cooperation with universities
- financial assistance and encouragement of governmental enterprises to invest in new software companies.

In the area of telecommunications equipment, Brazil extensively uses government procurement. The Ministry of Communication has a large research and development centre working on optical communications, switching and digital transmission. More than 60% of all national informatics industries depend on sales which are directly or indirectly linked to the Ministry of Communications, and as a consequence of the procurement policy, imported telecommunication equipment and components decreased from $245 million in 1975 to $54 million in 1978.[90]

6.2.2 Trans-Border Data Flows

In 1982 Brazil conducted several empirical studies focusing on policies towards trans-border data, information and software flows across Brazil's boundaries.

Trans-border data flows are extremely important in connection with computer activities, since they often result in a transfer outside of the country of information resources such as software, computer data bases, hardware and computer-related employment in general.

In June 1982 multinational subsidiaries in Brazil comprised 27 of 29 computer communication linkages outside the country, and the anticipated growth rate of these is three per month.

Brazil has four main objectives in dealing with information resources and trans-border data flows in particular:

- to enhance Brazil's political/cultural environment
- to maximize information resources located in Brazil, whether these are produced locally or are imported
- to maintain control over computer telecommunications technology
- to broaden the public access to information.

In implementing these objectives, Brazil has conceived policies governing telecommunications, telematics, informatics and trans-border data flows. In each of these areas, appropriate regulatory frameworks and policies have been instigated, and will be outlined in the following discussion.

The Brazilians make a distinction between two types of trans-border data flows: intercorporate flows -- those in which computer-based information and data is supporting of related activities; and commercial flows -- those in which data and information is itself the object of trade.

In the area of telematics, a national telecommunications holding company, Telebras, was set up in 1972 to digitize Brazil's public networks and to develop a national Brazilian technological capability. By 1980, Telebras had established the Sicram service, which is a computerized storage and retrieval system for messaging now operating through terminals which are linked to Embratel, Brazil's national telecommunications company, and also linked with Brazil's national telex network.

Brazilian telecommunications policy, which is focused on reduction of technology imports and on incentives for local production via Telebras' procurement, has positively affected the development of Brazilian technology. Industrial strategy in Brazil encourages the purchasing of equity in foreign informatics multinationals by Brazilian firms and also encourages the growth of new informatics industries. There is now considerable evidence that Brazil's telecommunications policy has actually stimulated the domestic telecommunications industry, e.g. in 1978, 90% of Telebras' acquisitions came from foreign affiliates, but by 1980 this figure had been reduced to 41%.[91]

The Brazilian informatics industry had its genesis in 1972 in the creation of CAPRE (The Coordinating Commission for Data Processing Activities). Given the job of creating an industrial policy for informatics, CAPRE had considerable powers to both restrict and regulate the importation, we have seen, of foreign data processing equipment, and also to protect the national Brazilian market from the unregulated importation of modems, minicomputers, perscoms and video terminals.

By 1979 SEI, the progeny of CAPRE, implemented a new strategy to upgrade the capability of national industries to produce more complex informatics technologies. As part of this policy:

> "Foreign affiliates are encouraged to exercise a comparative advantage and to produce advance state-of-the-art computer goods and

services for local consumption and export; they are also encouraged to improve local research and development facilities. Once a product can be manufactured by national capital, the respective market segments are protected to give the industry an opportunity to develop, while foreign affiliates are encouraged to shift toward more sophisticated products (instead of upgrading products in the same segment). The degree and type of protection is a function of the technological state of the products involved. As soon as international competitiveness is achieved, it is envisaged that protective barriers can be lowered, although measures may be considered necessary to ensure the continued improvement of local technologies and the permanence of national ownership of the country's informatics industries." (92)

Although they were initially infuriated by these policies, few multinationals left Brazil and most continued to prosper in the Brazilian market. Simultaneously, Brazil's share of this domestic informatics market increased to 36% by January 1982, with 71 private corporations (and one public corporation) producing more than 100 informatics products, predominantly for export. Locally produced data equipment, which comprised 30% of the total market in 1979, increased to 53% by 1981, and imports decreased from 29% in 1979 to less than 8% in 1981.[93]

SEI's policy efforts are now focused on microelectroncs hardware and software, and real time process control systems -- through the utilization of fiscal encouragement to set up specific types of manufacturing facilities, research and development centres, and increased use of government procurement and fiscal policies.

Brazil's first regulations on trans-border data flows in 1978 were a result of requests from MNEs to set up transnational computer communications facilities. At that time CAPRE ruled that these linkages would be approved for specific purposes only and for specific time periods. No restrictions were made on the content of actual messages, however. In the words of CAPRE, such a regulation was based on the principle that information and software are "economic resources, subject to trade and crucial socio-economic development". Brazilian policy in this area basically comprises a scrutiny of each potential computer telecommunications linkage and a determination of "to what extent individual applications for links fit with overall objectives."

Specific criteria which are applied involve whether the data flow is of a commercial or intercorporate nature, and what uses are being made of the information transferred to intercorporate computer communication systems, such as person-to-person communication, data processing, data base access, etc. The specific criteria for approval or disapproval of each link are summarized in Exhibit 6-20.

Since Brazil first instigated its policy on trans-border data flows in 1978 and 1982, there have been 32 applications for such linkages and seven refusals. Of the rejected seven, five involved access to data processing services located outside of Brazil, with the predominant reason for rejection involving the fact that the firms involved had made no effort to locate information resources inside Brazil. Approved and refused trans-border linkages, by type of user, are summarized in Exhibit 6-21.

Any data processing done outside of Brazil over computer communication linkages is absolutely forbidden.

Although it is too early to thoroughly evaluate Brazil's trans-border data flow policies, thus far there has been a definite increase in the number of both data bases, computers and software programmers in Brazil, and a general strengthening of Brazilian informatics industries has occurred without any significant damage to the amount of foreign investment.

EXHIBIT 6-20

Matrix of Brazilian Trans-Border Data Flow Policies

On-Line Use of Transborder Data Flows	Category of On-Line Transborder Data Flows: Corporate	Commercial
Data Communications	Person-to-person communications are not restricted	Brazilian PTT only; co-operation agreements possible
Data Base Access	Copy of data base in Brazil, whenever reasonable	Encouraged, but in co-operation with Brazilian institutions, preferably with copy of data base in Brazil. If no local copy, services are provided by the PTT, although co-operation agreements are possible
Data Processing (incuding use of software)	Not favoured	Not allowed abroad except in exceptional circumstances

Source: Special Secretariat of Informatics, 1983.

EXHIBIT 6-21

Transnational Computer-Communications Systems
Approved and Refused Applications by Type of User
March 1982

Type of User	Applications: Total	Approved	Refused	Pending
Government	2	2	-	-
Data Services	9	2	5	2
Transnational Corporations:				
Brazilian	4	4	0	-
Other	16	14	2	-
TOTAL	32	23	7	2

Source: Special Secretariat of Informatics, 1982.

6.2.3 Brazilian Software Policies

Realizing the importance of computer software compared to hardware, the Special Commission on Software has made the following policy recommendations.

Since the software and computer industries in Brazil are generally small to medium sized, they have great difficulty in obtaining development financing of software products. The SEI is cooperating with Brazilian funding agencies to make sure that there is sufficient financing available for both development and marketing of Brazilian-produced software, including funding for all phases of development, from research through start-up production. This funding also includes support of international marketing during the product's first year.

At the same time, realizing also that it is almost impossible to "establish realistic barriers to the physical importation of software produced in other countries",[94] Brazilian funding agencies have established a system in which software financing is provided for firms utilizing Brazilian technology and which insures that no exclusive use restrictions are being placed on the development of software. The Brazilians are also directing financing toward companies whose principal activities are software production.

In the area of international tenders, the Brazilians have drafted regulations which govern the importation of computer equipment and peripherals which are purchased with international financing. If computer equipment and systems are available from national producers they must be purchased inside Brazil, thus decreasing technological dependence and excessive imports.

In terms of software investment priorities, Brazil has given priority to software investments in which the country is least developed, i.e. basic business software.

The Brazilians are also actively attempting to increase software exports, which have already included software for banking terminals in Africa. Also in the area of internal fiscal incentives, the special commission recommended that the Ministry of Finance make available for up to ten years fiscal incentives according to which software users would be allowed a 200% deduction on any direct expenditures involved in the purchase of Brazilian developed software.

One of the most basic and controversial policies of SEI involves the registration of software. Noting that it is in the interest of Brazil to specify those types of software whose importation should be encouraged and to establish official prices for software to bolster the practice of separating software sales from those of computer equipment, the special commission on software has recommended that:

> "The SEI create a mechanism for registering all software products available in Brazil; that the SEI issue regulations instructing all government agencies and organizations operating either directly or indirectly under the federal administration to purchase only registered software products; that the SEI issue regulations which establish a method of making the granting of import licences on computer equipment dependent upn registration of the accompanying software; and that the SEI urge the Ministry of Finance to issue a regulation stating that the purchase price of software products may be included under operating expenses and will be tax deductible only in those cases where the software in question has been duly registered. (95)

Furthermore, with respect to the separation of software and hardware purchases, the Brazilians have realized that first of all it is in their interest to create a market for software products which is not dependent on foreign computer manufacturers, and also that it is absolutely necessary to establish some degree of competition between computer and software producers. The commission on software and computer services has thus recommended that the prices for hardware and computer software be clearly defined, and that in the case of computer imports, inclusion of the cost of the software be prohibited on the same import licence with equipment, and all software must be covered by separate contracts.

With respect to specific controls on basic software, the Ministry of National Revenue now requires that contracts with purchase of "basic software" are examined for the end of registration with INPI. The registration of software contracts with the INPI is the predominant means that Brazil uses to control purchases of foreign software, since each of these purchases always entails payments in foreign currency.

Finally, in the area of software training, the Brazilians have created new full degree programs in their universities, since there are no programs which are currently available which allow satisfactory levels of software training in Brazil, especially in the areas of basic software. The Brazilians are also encouraging teaching and research of software aimed especially at industrial automation processes.

The following section consists of SEI's summary software recommendations. Policy measures which, in modified form, may be applicable to Canada are stared (*).

6.2.4 Software Recommendations[96]

Recommendation No. 1: Funding of private software producers.

Whereas:

1. The software and computer services sector in Brazil consists of small and medium sized companies.

2. These companies have difficulty in obtaining financing for development of software products since they are unable to offer sufficient guarantees.

3. It is necessary to enhance the capacity of such companies to obtain funding.

4. It is difficult to establish realistic barriers to the physical importation of software produced in other countries.

5. It is necessary to assist software producers in obtaining access to their primary working instrument, in order to reduce direct production costs.

6. The production of these companies must be geared to equipment manufactured in Brazil.

It is recommended:

*1. That the SEI cooperate with official funding agencies to ensure that sufficient financing is provided for development and marketing of software produced in Brazil:

* a. Funding for all phases of development, from start-up to production.
* b. Funding for marketing the product during its first year.

2. That the SEI conduct studies aimed at creating an insurance or bond system to guarantee financing for development of software by Brazilian companies, under projects in which the SEI has an interest in the product to be developed.

3. That the SEI and the official funding agencies establish a system for ensuring that the software for which financing has been provided is produced using Brazilian technology only, thus reserving such funding for private companies interested in and capable of developing their capacity in this area.

*4. That in arranging such financing, preference is given to companies whose principal activity is the production of software.

*5. That the SEI cooperate with official funding agencies to ensure that financing is made available to software-houses for the purchase of Brazilian equipment, and that no exclusive-use restrictions are placed on the development of software, so that equipment may be purchased by firms acting as consortiums.

Recommendation No. 2: International tenders.

Whereas:

1. The rules for international tenders do not permit the reserving of markets or the protection of national firms when the purchase of equipment and information-system groups is subject to international financing.

2. The resources required for purchase of these information systems (including package groups) represents, in general, from 3% to 5% of the total package, but are nevertheless essential to the functioning of the group.

3. It is necessary to encourage users to purchase software developed by Brazilian companies.

It is recommended:

1. That the SEI cooperate with official funding agencies in giving priority to the development of Brazilian software for data processing to replace that purchased from foreign companies.

2. That the SEI draft regulations governing the importation of equipment and systems purchased with international financing, to the effect that where such equipment and systems are available from national producers, they must be purchased within Brazil, thus avoiding technological dependence and unnecessary imports.

*3. That the SEI create a funding program for the purchase of software developed to order by Brazilian companies.

Recommendation No. 3: Investment priorities.

Whereas:

1. Given the limited amount of resources available, the proper selection of priorities for optimum allocation of same is essential.

It is recommended:

1. That first priority be given to investments in those types of software in which Brazilian capacity is least developed: basic, support and applied software.

2. That preference be given to the development of software for equipment manufactured in Brazil.

Recommendation No. 4: Fiscal incentive.

Whereas:

1. It is essential that the cost of software produced in Brazil be reduced.

2. In the long run, it is a more efficient use of resources if users contract with specialized firms for the development of software, rather than depending on their own staff.

It is recommended:

*1. That the SEI urge the Ministry of Finance to make available for a period of ten years a fiscal incentive by which users would be allowed a 200%
Brazil.

Recommendation No. 5: Development of software in areas of national interest.

Whereas:

1. It is necessary to channel efforts in development and technical training into areas of strategic importance.

2. It is vital to strengthen national institutions which are developing software and therefore providing additional jobs for Brazilians.

3. High level research and development in software, while not necessarily leading to commercial applications, nevertheless contributes greatly to Brazil's level of technological independence.

It is recommended:

*1. That the SEI/DIGIBRAS promote the development, by Brazilian institutions, of products which are in the national interest and which lack commercial viability.

2. That the benefits of this development be extended to the community, under the direction of the SEI

3. That the SEI establish a program of fiscal incentives for companies with ongoing investments in the development of such products.

Recommendation No. 6: Exportation of software and computer services.

Whereas:

1. The ability of Brazilian companies in providing software and computer services on foreign markets is well-established.

2. Expanding the market for Brazilian companies which export computer services will improve their technological level and competitive position.

3. It is necessary for the computer sector to cooperate in equalizing its balance of payments.

It is recommended:

*1. That the SEI work in close cooperation with the MRE, the CONCEX and similar agencies, in order to facilitate and promote activities aimed at increasing exports of software and computer services.

Recommendation No. 7: Giving preference to Brazilian bids.

Whereas:

1. Private initiative is a fundamental characteristic of the socioeconomic model of Brazilian society.

2. It is desirable to expand potential markets for Brazilian producers of software services.

3. It is desirable to increase the competitiveness of the software industry within Brazil.

4. It is in Brazil's interest to reduce foreign dependence in those areas where the nation has the potential capacity to do so.

5. It is the responsibility of the public administration to promote growth of the computer sector under national control.

It is recommended:

1. That the SEI issue regulations instructing government agencies and state corporations to give preference to equipment and software services offered by Brazilian private industry, except for cases in which the material purchased is deemed necessary to the public interest or essential to national security, or in which its purchase within Brazil is deemed uneconomical.

2. That in cases in which software or consultation services are to be purchased abroad, and in which the SEI concludes that there are Brazilian products or services which meet the specifications and can be delivered by the time

they are actually required, the SEI should urge the purchasing companies involved to contract for such software services with Brazilian companies.

3. That the SEI should urge the Executive Office of the President of the Republic to issue a decree granting to the computer sector the same protection afforded to Brazilian engineering under Decree No. 64.345.

Recommendation No. 8: Property rights in the software sector.

Whereas:

1. It would be contrary to the interests of Brazil if existing provisions protecting individual property rights and copyrights were to apply to software.

It is recommended:

*1. That the computer sector itself be responsible for identifying cases in which the producer's rights have been violated and for pointing out any such irregulatities to the appropriate professional associations.

*2. That the SEI, the INPI and the appropriate professional associations study and formulate measures for preventing such violations and irregularities.

3. That the appropriate professional associations prepare, publish and promote a code of ethics covering conduct in the computer sector.

Recommendation No. 9: Registration of software.

Whereas:

1. It is necessary to create the means for publicizing the software products and services available in Brazil.

2. It is in the interest of the country to specify those types of software whose importation should be encouraged.

3. It is necessary to establish official prices for software products in order to strengthen the practice of separating the sale of software from that of computer equipment.

4. The creation of controls in the software sector will make it easier to establish subsidies for the protection of software producers.

It is recommended:

1. That the SEI create a mechanism for registering all software products available in Brazil.

2. That the SEI issue regulations instructing all government agencies and organizations operating either directly or indirectly under the federal administration, to purchase only registered software products.

3. That the SEI issue regulations which establish a method of making the granting of import licences on computer equipment dependent upon registration of the accompanying software.

4. That the SEI urge the Ministry of Finance to issue regulations stating that the purchase price of software products may be included under operating expenses and will be tax deductible only in those cases where the software in question has been duly registered.

5. That the SEI urge official funding agencies - the BNDE, FINEP, BACEN and BB (FIPEC) - to adopt the means for making the approval of any project which receives government financing, in whole or in part, and which involves the purchase of software products, subject to prior registration of the software in question.

6. That the SEI issue regulations stating that where it cannot be proven that a given software product was produced in Brazil, said product may not be registered until the INPI has studied and approved the service contract or transfer of technology agreement under which it was developed.

7. That the SEI/INPI study the characteristics of the software product (transfer of technology, copyright, industrial property), including with respect to foreign currency payments and rates.

8. That the SEI cooperate with the Central Bank to ensure that no payments are made in respect of copyrights on the software, including payments for manuals and required documentation.

Recommendation No. 10: Separation of hardware and software purchases.

Whereas:

1. It is in the national interest to create a market for software products which is not dependent upon computer manufacturers.

2. It is necessary to establish competition between computer manufacturers and software producers.

3. Purchases of computer equipment should not be allowed to serve as the means for introducing into Brazil software products whose transfer has not been approved by the competent bodies.

It is recommended:

*1. That prices for computer hardware and software be clearly defined, and that due account be taken of the sales price in the country of origin.

2. That the CACEX/SEI adopt internal provisions which, in the case of computer imports, prohibit the inclusion of the price of software or other

similar products and services on the same import licence with equipment. Software services must be covered by a separate contract.

3. That the SEI ask all interested parties to inform the CADE - pursuant to Article 2, (1) and (g), of Act No. 4137/62 ("Abuse of Economic Power") and Article 2, VIII, of Act No. 1521/51 ("People's Economy") - of any refusal on the part of computer manufacturers to negotiate for software services separately from the purchase of equipment or other products and services, or if prices are discounted in this respect.

Recommendation No. 11: Foreign currency payments.

Whereas:

1. It is essential that Brazilian producers of software services participate as instruments of support to the SEI.

2. It is necessary to improve the efficiency of mechanisms for controlling amounts paid in foreign currency for the purchase of software and computer services.

It is recommended:

1. That the SEI invite an appropriate body representing Brazilian producers of software and computer services to participate in the analysis of foreign contracts or agreements requiring payment in foreign currency for the purchase of computer systems, software and services.

2. That the SEI be given a role in controlling payment to foreign suppliers made by the INPI and the Central Bank for imported software products and computer services, and that emphasis be placed on the need to exercise this control at the product level (listing equipment, software and computer services), and not simply at the overall contract level.

Recommendation No. 12: Controls on basic software.

Whereas:

1. The registration of contracts with the INPI is one of the existing means for controlling the purchase of foreign software since such purchases entail payments in foreign currencies.

2. It is necessary to regulate the importation not only of applied and support software, but also of basic software.

3. The distinction between basic and other types of software is tenuous in many cases.

4. Interpretation Guidelines CST 79/75 of the Ministry of National Revenue does not at present include basic software.

It is recommended:

1. That the SEI urge the Ministry of National Revenue to amend Interpretative Guideline CST 79/75 to include the requirement that contracts for the purchase of "basic software" be examined for purposes of registration with the INPI.

Recommendation No. 13: Universities.

Whereas:

1. Technological advances are a direct result of providing adequate training for personnel and investing in research and development.

2. It is necessary that the training of personnel be coordinated with the effort to promote development and consolidation of a computer industry, which is entirely Brazilian in its basic features.

It is recommended:

*1. That the SEI promote the creation of a program to equip all universities (whether or not they now offer computer courses) with minicomputers of Brazilian manufacture, which are to be linked to teaching and research programs of university computer science and extension courses.

2. That the SEI be responsible for publishing the work, projects, models, theses, etc., that result from this program.

Recommendation No. 14: Training courses.

Whereas:

1. There is no program currently available offering satisfactory levels of training, particularly in the areas of basic software, support and firmware.

2. It is necessary to improve the efficiency of software production.

3. Disputes concerning the level of training within Brazil in the management of software production has served to inhibit the development of this industry.

It is recommended:

1. That a full degree program in computer engineering be created to provide Brazil with duly trained personnel.

2. That the above-mentioned full degree program be drawn up in cooperation with the CREA, ABENGE and MEC

*3. That the SEI determine the means necessary to ensure that Brazil obtains personnel with sufficient training in management of software production.

*4. That the SEI develop a publication and feedback plan for this training program.

Recommendation No. 15: Improvement of existing courses.

Whereas:

1. It is necessary to ensure that the number of trained personnel does not exceed demand.

2. It is essential to promote constant upgrading of existing courses.

It is recommended:

*1. That the MEC regulate technical training courses for domestic and foreign production of administrative software, evaluating the quality of these courses, assessing their relationship to the job market, and creating requirements in terms of minimum standards that must be met.

*2. That present graduate study programs be maintained and consolidated.

*3. That professional development courses be promoted as a means of supplying short-term personnel needs, through the offering of graduate programs and specialized extension courses.

*4. That the SEI cooperate with professional associations in evaluating their course offerings with a view to bringing these up to satisfactory levels.

Recommendation No. 16: Software for other technological sectors.

Whereas:

1. The development of software is an activity with applications in various areas of knowledge.

2. It is necessary to substantially increase technical training in the production of software for industrial automation.

It is recommended:

*1. That academic institutions promote or encourage teaching and research in the types of software used in other technological sectors.

*2. That teaching and research in software aimed at the automation process be encouraged.

*3. That the SEI cooperate with the institutions employing automation in setting standards and promoting the development of their projects within Brazil.

<u>Recommendation No. 17</u>: Contract guarantees.

Whereas:

1. It is necessary to promote a relationship based on rationality and fairness between users and suppliers of software and computer services.

It is recommended:

1. That the SEI promote the formulation of model articles for software and computer service contracts.

2. That the SEI examine the possibility of stipulating that, in the process of registering imported software, suppliers be required to furnish proof of delivery of the open group package to a recognized receiver.

<u>Recommendation No. 18</u>: Standardization.

Whereas:

1. Standardization is an important tool which may be used in expanding the market for Brazilian products.

2. On the other hand, such standardization ought not to restrict the activities of Brazilian producers.

It is recommended:

1. That the SEI cooperate with Brazilian manufacturers of computer equipment and the CB-21 of the ABNT, in drawing up specifications for standardizing programming languages and inherent terminology.

2. That the ABNT draw up the minimum requirements that languages used by Brazilian manufacturers must meet, based on programming languages in use at the world level.

3. That it is not necessary to require a single standard programming language for all equipment built in Brazil.

<u>Recommendation No. 19</u>: Publications.

Whereas:

1. There is an almost total lack of knowledge regarding studies, research and theses on software carried out in Brazil, as well as regarding the commercial viability of products in this area.

It is recommended:

*1. That the SEI publish (or promote publication of) catalogues on personnel training.

*2. That the SEI publish (or promote publication of) a catalogue of Brazilian software products and computer equipment.

*3. That the SEI publish (or promote publication of) a catalogue of Brazilian firms, together with their products and areas of interest.

*4. That the SEI promote publication of a technical journal containing papers and national bibliographic studies of work done in the sector.

*5. That Brazilian and foreign firms provide the SEI, at regular intervals, with a list of clients and software provided under licence, so that the SEI can maintain statistics regarding same.

<u>Recommendation No. 20</u>: Exchange of software.

Whereas:

1. It is necessary to ensure optimal use of resources allocated for development of software.

It is recommended:

*1. That the SEI promote the exchange of software developed by government agencies and universities.

<u>Recommendation No. 21</u>: Obtaining funding for computer activities.

Whereas:

1. The Fund for Advancement of the Information Sciences (FAI) will constitute one of the most important instruments for promoting development within the sector.

2. It is appropriate that the computer sector itself should provide the FAI with additional resources to augment the funding it receives from the government.

It is recommended:

1. That the SEI urge the Ministry of Finance to establish a mechanism by which financial resources may be reserved for the computer sector (similar to methods now used for the FINOR, SUDAN, etc.).

2. That the SEI ask the Ministry of Finance to issue regulations which will remain in effect for a stipulated amount of time, reserving a given percentage of the income tax due on profits earned by computer companies, and specifying that this amount be paid by the company to the FAI, while the remaining amount is collected in the normal taxation process.

Having reviewed Brazil's protective policies toward software development, we now turn to the software policies of a contrasting country, Singapore.

6.3 Singapore's Software Strategy

Although the newly-industrialized countries (NICs) of Southeast Asia -- Singapore, Thailand, Indonesia, Malaysia and the Philippines -- had extremely high growth rates during the 70s, the recent global recession has cut these growth rates in half, and the countries have been further squeezed by significant drops in oil revenues. Consequently, many nationalistic measures prevalent in the 70s have been dropped, and most of these countries are now calling for more foreign investment. In a major policy shift, the Asian Development Bank is now intending to provide equity financing to private firms in selected Asean countries. One of these is Singapore, a small country at the tip of the Malay Peninsula with few natural resources, save the ingenuity of her 2.5 million people. With the constraints of this limited manpower resource, Singapore has recently conceived and implemented an industrial strategy which involves the development of high value-added technology and knowledge-based industries -- such as software, computer service industries, automated banking and finance, engineering and medical/scientific consultancy services.

This industrial strategy is comprised of three sets of inter-linked measures which will be described in this section -- development support measures directed at these new industries, export support measures and manpower training/upgrading.

One of the main thrusts of Singapore's new industrial development strategy is manpower development to train workers to meet demands of the high technology industries and "brain" services which Singapore is promoting. Here the Economic Development Board has supplemented other government training efforts through the operations of its own special training centres for technology, which are collaborative efforts with the private sector and other governments. These are: The Japan-Singapore Institute of Software Technology, the German-Singapore Institute of Production Technology and The French-Singapore Institute of Electro-technology.

As of March 1982, 2,298 apprentices and technicians had completed two years training programs in these institutes and had been placed in 215 companies, while 1,242 were still in the process of being trained.[97]

In total, these training institutes have covered specialized areas such as computer-based tool and die design, computer control of production processes, robotics, microprocessor and computer applications, and software technology.

6.3.1 Recent History

In March 1980 a ministerial Committee on National Computerization was put into place by the Singapore government to create a strategic plan with the aim of, "establishing Singapore as a centre of computer services and software." This committee, chaired by the Minister for Trade and Industry, was given the task of creating a master plan to build up microelectronics expertise in Singapore and to create an export-oriented software industry.

Specific recommendations of this report involved:

- promoting the computer software industry
- increasing the number of computer professionals
- accelerating computerization in the civil service.

One recommendation also involved the establishment of a National Computer Board (NCB) which would implement this plan for the development of computer software.

In addition to promoting the computer service industry, the National Computer Board has formulated a series of innovative programs and industrial incentives to induce firms to set up software development companies in Singapore, particularly those which deal with software developed for state-of-the-art technology. Also, to answer the shortages of software professionals, a series of programs has been instigated to increase and improve the training of data processing professionals, and these programs are collectively producing a supply of approximately 1,000 information systems professionals yearly.

6.3.2 Computer Education and Training

In its preliminary 1980 report, the Committee on National Computerization had identified the absence of trained software and computer professionals as the "main barrier" to further developing Singapore into a centre of excellence for computer services. Thus, one of the Committee's main recommendations asserted that efforts should be placed into quickly training computer professionals. Since the Singaporeans initially estimated that the number of software writers necessary for the computer service industry would be between 5,800 and 7,800 by 1990, and their their current number of professionals in this area was around 1,200 in 1980, they estimated a need to increase the software writing pool by five to six times within eight years. On a yearly schedule, they have kept to this ratio thus far.[98]

To satisfy this demand for computer professionals, Singapore has instigated a plan involving the following elements:

Before the 1980 formation of the Committee on National Computerization, Singapore's software and computer professionals came from in-house training programs of multinational computer vendors and firms widely using computers, such as national banks and insurance organizations. The government first instigated financial/export incentives to upgrade and expand the quality of these multinational training programs and also established computer-related training institutes.

These efforts were correlated with upgradings and expansions of computer programs throughout the school system, and involved the creation of the Institute of System Science (a collaborative effort between the National University of Singapore and IBM), the Japan-Singapore Institute of Software Technology, the French-Singapore Institute of Electrotechnology, the German-Singapore Institute of Production Technology and the Centre of Computer Studies.

The Institute of Systems Science is a post-graduate institute instigated in October 1981 with IBM, which trains around 100 systems analysts yearly. It is also providing courses in training senior executives in Singapore to use computers.

The Centre of Computer Sciences is a new institute which trains around 200 programmers yearly at the polytechnic level. Begun in 1983, it is a joint collaborative effort between the British Council and the Ngee Ann Polytechnic.

The German-Singapore Institute of Production Technology began operations in February 1982, and trains production engineering technicians and vocational instructors who specialize in computerized machining processes.

The Japan-Singapore Institute of Software Technology (JSIST), which also began operations in February 1982, provides training in computer software. Its purpose is to increase the supply of computer personnel requisite for the national computerization program as well as for the development of the computer software industry in Singapore. The Japanese-Singapore Institute is also upgrading EDP managers and is providing middle and senior management training in the applications of computers. They now have an output of around 300 persons per year. This project was officially launched with the signing of the Record of Discussions in Singapore in December 1980. With this agreement the Japanese government (i.e. NEC) agreed to provide technical assistance to Singapore in the form of NEC computers and a team of six software experts for five years.

Although it is thought that Japan's expertise in alphaphotographic Videotex and facsimile transmission will enable her to be one of the first to penetrate the Chinese market, Japan is not known for software expertise, and this cooperative venture with Singapore must be seen as the first real effort to develop software markets in Southeast Asia.

The French-Singapore Institute of Electrotechnology began operations in April 1983 and will specialize in training for electronic manufacturing and servicing industries. Main emphasis here has been placed in the area of robotics, microprocessors and computer applications.

All of these manpower training institutes are a creation of Singapore's Economic Development Board, which as we have seen gave high priority in 1981 to ensuring that an adequate supply of highly skilled persons would be available for the computerized industrial sector.

In early February 1984 the Sperry Corporation announced that they would open a software development centre in Singapore within the year. An extension of Sperry's applications development centre in West Germany, the Singapore centre will be the only location outside of Germany which will handle the development and maintenance of Unidas software. (Unidas is the document retrieval system of Sperry and has been used worldwide in a variety of installations).

The Unidas software development centre will concentrate on:

1. further development of Unidas software, including specific enhancements for regional customers and AI developments.

2. Unidas support and maintenance of thesaurus applications and specialized technical support for Sperry's Unidas customers throughout Southeast Asia.[99]

In addition to the above institutes for training, in 1983-84 Singapore set up the following three specialized training units: (1) the Computervision CAD/CAM training unit, set up in partnership with Computervision Corp. of the US and operational since February 1983. This unit presents application oriented courses in subjects such as printed circuitboard design and numerical control. (2) the Singaporeans have recently set up the Asza/Economic Development Board Robotics Training Unit (a partnership with Asza of Sweden)[100] which began operations in June 1983 and (3) a new cooperative training group with Japax, Ikegai and Hamai -- all of Japan.

It is expected that the total output of computer professionals in Singapore will soon be around 700 to 1,000 per year.[101]

In implementing the new computer software policies, Singapore is acting very quickly and it has achieved most of its stated goals already. Because it is a small country, once policy has been formulated Singapore may respond very quickly without the need, as in Canada, for endless task forces and field trials of new technology. Quite simply, Singapore is attempting to guarantee that training of its software writers and computer professionals will be able to respond to market demands as Singapore implements its industrial strategy to transform itself into an area of software expertise for Southeast Asia.

6.3.3 Recent Industry Statistics

In late 1982 Singapore's National Computer Board carried out two surveys of its computer industry.[102] The main results of these surveys are as follows:

> Computer users are anticipated to double from 1,800 in 1982 to 3,600 in 1985, and Singapore's private sector comprises 94% of the computer user base, compared with 6% in the public sector.
>
> Most of the applications software running on perscoms in Singapore is used for finance, administration and distribution functions. Few business establishments are using modelling tools or simulations to predict profitability.
>
> More than half of the software packages identified in the survey were written in Cobal. The survey also showed that half of all software packages are custom made and that only 30% are standard packages.
>
> Most of these software packages (two-thirds) were developed in-house. However, as in other countries, the use of standard packaged software is increasing in Singapore.
>
> At the time of the survey, Singaporean computer users spent on average only $1 on software for every $10 on hardware purchases; however, this percentage is also changing rapidly.
>
> With respect to manpower, there are presently around 2,800 computer

professionals in Singapore, two-thirds of whom are systems analysts or programmers.

Sales of software services and applications packages in Singapore rose from $2 million in 1977 to more than $30 million in 1981, and exports of software services and packages increased from around $500,000 in 1977 to more than $7 million in 1981.

Although today there has been a startling rise in Singaporean computer manpower, most of these professionals have less than two or three years of data processing experience.

6.3.4 Industrial and Export Support

Fiscal, tax and export incentives have been actively used by Singapore to implement a policy of restructuring industry toward more technology-intensive activities which are higher value added. Progressive tax incentives are given to firms involved in technology upgrading, automation and software training projects. There are, for example, no import taxes on software packages.

These are the major incentives offered by the Economic Development Board to industry:

1. Export Incentives:

   These incentives give a 90% tax exemption on all profits above a specified base which are a result of export sales. The incentive is granted for three or five years, with the longer period being granted to companies which have not received a Pioneer Status Incentive.

2. Pioneer Status Incentive:

   The Pioneer Status allows for total tax exemptions of 40% of a corporation's income tax for five to ten years, with this exemption being granted, amongst others, to firms which engaged in sophisticated software development and export. This has been the most extensively used incentive by software firms.

3. Investment Allowance:

This is an incentive for both manufacturers and technical service companies. It is considered as an alternative to the above two incentives.

In Singapore the investment allowance has mainly been used to promote automation. Under this incentive a firm is given tax exemptions on, "a specified amount of profits equal to the approved investment allowance which is a percentage (up to a maximum of 50%) of the fixed investment in plant, machinery and factory buildings actually incurred by the company on the project."[103]

Such allowances are granted as a bonus over the normal capital allowances given to a firm. Thus, any firm can at the same time claim a three year accelerated depreciation on plant and any machinery and not have any benefits of the investment allowance reduced.

4. The International Consultancy Services Incentive:

The incentive was instigated to increase the number of software, consultancy and engineering firms located in Singapore which export. It gives a five year 20% tax write-off on export profits which are above a specified level. Export services which are eligible for the incentive comprise software, design and engineering, machinery pro duction, data processing and all technical advisory services. However, only companies which have a gross export income of at least $1 million are eligible. To date no software firms have been granted this incentive because (a) no firms have had a million dollars in software exports from Singapore, and (b) there has been difficulty with consulting firms in determining what percentage of exports is due to consulting and what percentage is due to software production.

5. <u>Tax Incentives for Research and Development:</u>

   In addition to the above incentives, Singapore has introduced the following R&D incentives:

   a. a double deduction of 200% of any R&D expenditures, except those on buildings and equipment

   b. accelerated depreciation over three years for machinery and plants used in R&D

   c. investment allowances of a total of 50% of R&D capital investment, with the exception of building costs

   d. capitalization of lump sum payments for manufacturing licences for five years.

In addition, the Singapore Science Park, which is situated next to the National University of Singapore, has recently been developed to accommodate R&D organizations and activities in connection with the manufacture of high technology products. As part of this science park, the National Computer Board will operate a Software Technology Centre. Although their R & D incentives have been in place only a short time, thus far little original R & D is being done in Singapore.

The central features of various tax incentive schemes applicable to software are summarized in Exhibit 6-22.

EXHIBIT 6-22

Incentives for Investors

Tax Incentives	Qualifying Activites	Statutory Conditions	Tax Concession
• Pioneer Status	Approved manufacturing and related activities. ·knowledge-based services.,	No statutory conditions but the EDB generally imposes fixed investment conditions to be achieved by 4th or 5th year of production.	1. Exemption of 40% tax on profits arising from pioneer activity. 2. Tax relief period is 5-10 years.
• Expansion Incentive	As above.	Minimum investment of S$10 million in productive equipment and machinery.	1. Exemption of 40% tax on profits in excess of pre-expansion level. 2 Tax relief period of up to 5 years.
• Export Incentive	As above and deepsea fishing products.	EDB generally imposes fixed asset investment conditions to be achieved.	1. 90% exemption of tax on export profits in excess of a specified base. 2. Tax relief period is 3 years for pioneer companies and 5 years for non-pioneer companies.
• Investment Allowance Incentive	1. Manufacturing. 2. Specialised engineering & technical services. 3. Research and development activities. 4. Construction. (The incentive is increasingly being used to promote the upgrading and mechanisation of existing operations.)	Qualifying period of up to 5 years within which investments must be made.	Tax deduction equal to a specified proportion (up to 50%) of new fixed fixed investment.
• Warehousing and Servicing Incentive	Warehousing, technical or engineering services. (The incentive aims at encouraging warehousing of engineering or other technical products as a stepping stone to manufacturing.)	1. Minimum fixed investment in warehouse buildings and equipment of S$2 million. 2. Goods must be traded by company.	1. 50% exemption of tax on profits in excess of a fixed base. 2. Tax relief period of 5 years.
• International Consultancy Services Incentive	Consultancy services in respect of overseas projects. These include technical assistance, design, engineering, fabrication of equipment and data processing.	Minimum revenue of S$1 million per annum.	1. 50% exemption of tax on profits in excess of a fixed base. 2. Tax relief period is 5 years.
• Approved Foreign Loan Scheme	Manufacturing and related activities.	1. Minimum loan of S$200,000 from a foreign lender for purchase of productive equipment. 2. Tax relief should not result in an increase in tax liability in the foreign country.	Exemption of withholding tax on interest.
• Approved Royalties	Manufacturing and related activities.	Tax relief should not result in an increase in tax liability in the foreign country.	50% or 100% exemption of withholding tax on royalties.

The Capital Assistance Scheme

In addition, Singapore has a variety of financial assistance schemes available through industry, the first being the Capital Assistance Scheme. The CAS is a tool to promote projects of strategic value to industrial development in Singapore including software development. Under the CAS, projects can obtain fixed interest loans at very low interest rates for long terms. The scheme basically acts as a catalyst to establish desirable industries through providing equity capital. By March 31, 1983, a total of S$157.6 million had been committed to establish or expand 20 projects including software, aircraft components, photomasks for wafer diffusion, mainframe computer systems, etc.[104]

The Small Industry's Finance Scheme

The SIFS attempts to encourage small local industries, firms with fixed assets of no more than S$2 million at the time of application. Realizing that small companies such as software companies may provide vital support to larger firms, the SIFS provides fixed cost funds to participating banks for further lending to small local manufacturing or high technology service enterprises. These funds are jointly administered by Singapore's Economic Development Board and participating banks. Again, as of March 31, 1983, S$247.96 million had been approved for more than 900 loans.[105]

The Skills Development Fund

Focusing on equipping employees with new skills and knowledge, the Skills Development Fund, established in October 1979, provides financial assistance to firms for the training of skills critical to Singapore's restructuring effort. This fund, financed through a levy on employers, involves 4% of the salaries of all employees who earn less than S$750 monthly. There are two types of financial incentives offered under the Skills Development Fund.

First, the skills training scheme provides direct grants to employers for existing training programs to undertake upgrading of worker skills. This scheme may cover

30%-90% of allowable training costs, and priority development is given to technical skills, computer and software related skills, management and supervisory skills, product design and R&D skills, and basic education for skills training projects.

The second scheme, the interest grant for mechanization scheme, introduced in December 1980, encourages firms to invest in new automation machinery and equipment including computers to eliminate labour intensive methods. Companies purchasing or renting machinery and equipment under this scheme can apply for a grant to reduce the interest cost. The interest grant comprises half of the actual interest which has occurred in financing equipment or machinery purchase, with a maximum of 9% yearly, as calculated bi-annually. (In 1983 the Singaporeans widened the scope of this scheme to allow assistance in financial leasing for automated machinery and to provide full equipment subsidy for priority Singaporean support industries, which themselves may have obtained equipment loans under the smaller industry finance scheme).

As of March 1983, training grants have received 11,998 applications, and total funds committed involved S$120.77 million. The interest grant for mechanization received 467 applications with a total committment of S$25.18 million, and a third incentive, the development consultancy scheme, introduced in 1981 to provide financial assistance to local firms which engaged external consultants in short term high priority projects, has received 94 applications for a total grant of only S$4.39 million.

Total grants which have been committed under the training grant scheme involved training of almost 100,000 workers in new skills.

This feat is roughly equivalent to training a million Canadians in new skills in a three year period.

6.3.5 Summary

Heavily implicated in computer and software industries, Singapore has discouraged labour-intensive factories by increasing wages. In 1981, the government controlled pay scale increased 20%. (Having long restrained wages, Singapore ended up with a shortage of labour which was filled by 30,000 Asian "guest workers", and the present strategy of higher wages has not decreased foreign investment, which now provides nearly 73% of all investment in plants and machinery.)

The Japanese have been invited, we saw, by the government to help develop Singapore's EDP user community, and the recently established Japan-Singapore Institute of Software Technology is now training senior level systems analysts and programmers. The Republic has also committed more than $100 million to computerize its ten ministries over the next decade.

An equivalent computerization effort is occurring in the commerce, banking and financial sectors, and one international consulting firm projects that Singapore will acquire more than 1,000 minicomputers in 1983 alone and that this number will double in 1984 and 1985.

Over the past decade Singapore has really graduated from an assembler of consumer products to an integrated manufacturer of advanced consumer electronics. In mid-1979 it branched out also into manufacturing high grade components for industrial electronic equipment. The new policy of upgrading artificially suppressed wages is forcing many manufacturers to automate and also to diversify into more sophisticated electronic products.

Government policy encourages industry to specialize in three main areas: high quality components such as semi-conductors and microwave components; industrial electronics such as avionics equipment, electronic measuring equipment and computer peripherals; and software. Principal statistics of the Singapore electronics industry, the industry's structure and exports of major electronics equipment are shown in Exhibits 6-23 through 6-25.

EXHIBIT 6-23

Principal Statistics of the Electronics Industry*

Year	No. of Establish-ments	Output (S$ million)	Employment	Value Added Per Worker (S$)
1968	5	8	700	2,000
1969	15	67	7,500	6,600
1970	35	213	11,250	8,810
1971	49	319	15,870	8,890
1972	53	617	27,270	10,500
1973	64	1,097	39,210	10,830
1974	91	1,603	46,230	11,300
1975	95	1,458	32,030	14,800
1976	105	1,988	43,720	14,600
1977	117	2,323	46,440	15,200
1978	135	2,822	53,440	16,700
1979	168	4,093	66,840	19,100

* Source: Singapore Business, November 1981, p.51.

EXHIBIT 6-24

Structure of the Electronics Industry

	1974	1977	1980	1990
Industrial Electronics	4%	2%	3%	20%
Consumer Electronics	41%	48%	40%	35%
Electronic Components	55%	50%	57%	45%

EXHIBIT 6-25

Singapore Exports of Major Electronic Products
($ million)

Year	All Exports	Domestic Exports
1979	4,397.3	1,709.6
1978	2,944.2	1,314.8
1977	2,238.3	1,065.5
1976	1,761.6	748.7
1975	1,13.0	471.3

Source: Singapore Business, November 1981, p.51.

Finally in early 1984 a Chinese firm, Guangzhou Audio and Appliance Factory, set up a 50/50 percent joint venture with the Lityan Group of Singapore[106] to manufacture perscoms in China. Under a five year pact, this Singapore firm will jointly develop, assemble and market microcomputers in China. Projected Chinese sales for a five year period are in the range of $80 to $90 million for hardware and over triple this in software spinoffs. Singaporeans will provide components, test equipment and technical expertise. The Chinese will provide factory assembly space and power facilities.

6.3.6 Export Retaliation

A number of US companies are putting pressure now on Singapore to sign the subsidies code of the General Agreements on Tariffs and Trade (GATT), and the US has threatened to impose counterveiling duties upon its exports if Singapore does not sign. On the other hand if Singapore does sign, it might be forced to end many export incentives which have worked well in that country.

It all started with compressors. In mid-1983 the US commerce department ruled that Masushita of Singapore had been receiving subsidies from the Japanese government which were equivalent to 5.7% of the value of these exports. At the same time, American firms in the compressor business complained to Congress that the government of Singapore had been unfairly subsidizing the Singapore subsidiary of Masushita's, which was exporting compressors to the US.

To delay the imposing of counterveiling duties against Masushita, the Singaporean government voluntarily decided to impose export levies of this amount (5.7%) on the compressors in question.

Although other Asian NICs such as South Korea, Taiwan and Hong Kong have already signed this, or similar, agreements with the US, Asean nations such as Indonesia, and Malaysia have not signed.

A number of Asian countries have removed direct export subsidies in compliance with GATT, but they still indirectly support their exporters. In the specific case of the compressors, the Commerce Department had said that Singapore's reduced tax rate on export profits in selected industries was a form of indirect subsidy. It also ruled as unfair Singaporean practices of rediscounted export bills of exchange and even attacked retraining grants.

6.4 The United States

6.4.1 Software Suppliers, Markets and Growth

The US is the main market for, and supplier of, software in the world. The leading US software suppliers are shown in Exhibit 6-26, and leading independent suppliers in Exhibit 6-27.

Estimated growth figures for the US software market are presented in Exhibit 6-28. The total US software market, as estimated by Input Inc., was $4.7 billion in 1979 and $6.1 billion in 1980, an annual growth rate of 30%. Sales of applications software took 19% and 18% of total software sales in 1979 and 1980, and the equivalent figures for systems software were 23% and 25%.

Perscom sales increased 300% in the first quarter of 1984, and it is predicted that sales of systems software will show average annual growth rates of 27% in the US, with systems software revenues increasing from $1.1 billion in 1979 to more than $15 billion by 1990. This will be a change in total market revenues of 23% in 1979 to 34% by 1990.

US applications packages, with an expected annual growth rate of 30% to 33%, will increase sales from approximately $1 billion in 1979 to $17 billion in 1990, a market share increase from 19% to 39% during this period.

As in Canada, custom software is expected to die a slow death, with the exception of the customization of applications packages. The reasons for this trend involve mainly the cost ratio of packaged vs in-house development, which can vary from 1:50 for systems software for mainframe computers to 1:3,000 for applications software for perscoms.

EXHIBIT 6-26
Leading Software Suppliers in the US

Rank 1982	Rank 1981	Company	Software Revenues ($ Millions) 1982	1981	1980	Total Corp Revenues ($ Millions) 1982	1981	1980	Fiscal Year Ending	No. of Employees	Revenue per Employee
1	1	IBM Corp	2,405	2,034	1,878	36,364	29,070	26,213	12/31	364,796	94,201
2	2	Control Data Corp	1,202	1,178	1,036	4,292	4,163	3,784	12/31	56,000	76,643
3	3	Digital Equip Corp	1,887	814	589	3,881	3,198	2,368	06/30	68,000	57,074
4	5	Burroughs Corp	795	613	580	4,168	3,405	2,902	12/31	62,000	67,516
5	4	NCR Corp	740	687	598	3,326	3,433	3,322	12/31	65,000	54,246
6	7	Automatic Data Proc.	669	558	455	669	558	455	06/30	15,000	44,617
7	6	Computer Sciences	630	601	453	630	601	453	03/31	14,000	45,012
8	8	TRW Inc	625	535	425	5,132	5,300	4,984	12/31	85,000	60,376
9	9	EDS Corp	482	455	375	510	455	375	06/30	11,330	45,013
10	12	General Electric Info	476	350	222	600	500	327	12/31	5,500	109,091
11	10	Sperry Corp	420	432	383	5,243	5,427	4,785	03/31	78,581	66,721
12	11	Hewlett-Packard Co	412	352	305	4,200	3,578	3,098	10/31	64,000	65,625
13	14	McDonnell Douglas	328	226	179	328	226	179	12/31	5,890	55,688
14	13	Tymshare Inc	250	243	217	297	290	236	12/31	3,500	81,857
15	16	Harris Corp	228	209	172	1,719	1,551	1,301	06/30	26,000	66,115
16	15	Honeywell Inc	220	214	205	5,490	5,351	4,925	12/31	94,000	58,404
17	19	Perkin-Elmer Corp	211	156	141	1,037	1,116	996	07/31	14,100	73,546
18	17	General Instrument	210	181	158	957	825	718	02/28	26,500	36,113
19	24	Boeing Computer Serv	171	132	90	267	206	140	12/31	7,000	38,142
20	20	Informatics General	170	150	126	170	150	126	12/31	2,600	65,449
21	22	National CSS Inc	168	146	121	670	587	502	12/31	1,000	15,500
22	25	Shared Medical Sys	166	132	107	166	132	107	12/31	1,800	22,096
23	18	Allied Info Systems	150	158	147	489	492	460	12/31	9,000	32,111
24	23	United Telecomm Comp	149	160	152	149	160	152	12/31	1,684	88,242
25	21	University Computing	141	147	118	141	147	118	12/31	1,845	75,152

Source: ICP Software Business Review, June/July 1983.

EXHIBIT 6-27

Leading Independent Software Suppliers

No.	Company	1982 Rank	Total Corp Revenues ($ millions)	Percent Software Revenues	Total Software Revenues ($ millions)
1	MSA Inc	33	101	100	101
2	Informatics General	20	170	40	68
3	Applied Data Research	41	68	85	58
4	University Computing	25	141	37	52
5	Cullinet	50	49	100	49
6	Computer Assoc. Int.	57	43	100	43
7	Cincom Systems Inc.	54	44	93	41
8	McCormack & Dodge	65	38	100	38
9	American Mgmt. Sys.	39	70	51	36
10	Pansophic Systems Inc.	69	36	100	36
11	Microsoft	85	26	100	26
12	Anacomp Inc.	47	110	22	24
13	MicroPro Int. Corp.	94	24	100	24
14	Software AG of NA Inc.	92	25	100	24
15	Digital Research Inc.	96	23	100	23
16	Comserv Corp.	86	25	88	22
17	Policy Mgmt. Sys.	55	44	50	22
18	Kirchman Corp.	61	41	51	21
19	Continuum Co.	108	20	100	20
20	Mgmt. Decision Sys.	88	25	72	18
21	Information Builders	109	20	85	17
22	American Software Inc.	115	16	100	16
23	Boole & Babbage Inc.	118	16	100	16
24	Rand Info. Systems	117	16	83	14
25	EPS International	122	14	93	13

Source: ICP Software Business Review, June/July 1983.

EXHIBIT 6-28

US Software Industry Forecast to 1990

Software Type	1979	1980	1981	1982	1983	1984
SYSTEM SOFTWARE						
- $Millions	1,100.00	1,500.00	2,009.40	2,632.91	3,433.56	4,458.96
- %Growth		36.4	34.0	31.0	30.4	29.9
- Percentage of Total Market	23	25	27	29	31	33
APPLICATION SOFTWARE						
- $Millions	900.00	1,100.00	1,637.24	2,360.74	3,322.80	4,594.08
- %Growth		22.2	48.8	44.2	40.8	38.3
- Percentage of Total Market	19	18	22	26	30	34
SYSTEM DEVELOPMENT						
- $Million (1)	2,700.00	3,500.00	3,795.42	4,085.55	4,319.64	4,458.96
- %Growth		29.6	8.4	7.1	5.7	3.2
- Percentage of Total Market	57	57	51	45	39	33
TOTAL	4,700.00	6,100.00	7,442.00	9,079.00	11,076.00	13,512.00
% Growth		30	22	22	22	22

	1985	1986	1987	1988	1989	1990	Average Annual Growth 1979-1990
SYSTEM SOFTWARE							
- $Millions	5,796.40	6,837.40	8,341.56	10,176.88	12,415.78	15,147.34	
- %Growth	29.4	18.5	21.9	22.0	22.0	22.0	27.0%
- Percentage of Total Market	35	34	34	34	34	34	
APPLICATION SOFTWARE							
- $Millions	5,769.40	7,038.50	8,832.24	11,074.84	13,876.46	17,374.89	
- %Growth	25.6	22.0	25.5	25.4	25.3	25.2	31.2%
- Percentage of Total Market	35	35	36	37	38	39	
SYSTEM DEVELOPMENT							
- $Million (1)	4,945.20	6,234.10	7,360.20	8,680.28	10,224.76	12,028.77	
- %Growth	10.9	26.1	18.1	17.9	17.8	17.6	14.8%
- Percentage of Total Market	30	31	30	29	28	27	
TOTAL	16,484.00	20,110.00	24,534.00	29,932.00	36,517.00	44,551.00	
% Growth	22.0	22.0	22.0	22	22	22	22.7%

Source: Extrapolated from data provided by Input and Hambrecht & Quist (does not include bundled software revenues).

(1) Includes consulting and educational services to upgrade or enhance clients electronic processing staffs as well as custom developed software.

Microsoftware is expected to grow annually at 50%, as compared to 39% and 31% growth rates for minisoftware and mainframe software, respectively.

6.4.2 High Tech Co-Ops

In response to Japan Inc., US companies are joining into high tech co-ops to fight Japanese government-subsidized exports in computers, semi-conductors and software.

Two new US research cooperatives have been set up -- the Microelectronics and Computer Technology Corp. (MCC) and the Semi-Conductor Research Corp. (SRC). The former, located in Austin, Texas, has about 50 researchers and by the end of 1984 expects to increase that number by five times with a budget of $75 million. The second organization has committed $8.25 million to support research projects in 30 US universities[107] and is planning to devote $100 million in a half-decade project to develop memory chips.

MCC is performing its own computer and software R&D, and a number of firms will contribute the 250 researchers needed by the end of 1985. However, the entry fee is $200,000 for participation, and members must agree to share expenses for three years. Only participating firms are entitled to the results and benefits of any project's research.

These are the priority research areas of MCC:

1. Alpha-Omega - MCC's response to the fifth generation computer project. This will receive most of MCC's funding -- $50 million per year for a decade and will involve studies of new parallel processing computers, models for data bases and studies in artificial intelligence.

2. Software - with a budgeting level of $8 million per year for eight years, the Software project will create programs for hardware developed in Alpha-Omega.

EXHIBIT 6-29

Consortium Members of the Microelectronics and Computer Technology Corporation

Advance Micro Devices Inc.
Control Data Corp.
Digital Equipment Corp.
Harris Corp.
Honeywell Inc.
Martin Marietta
Mostek
Motorola Corp.
National Semiconductor Corp.
National Cash Register
RCA Corp.
Sperry Corp.
United Technologies

3. CAD/CAM - MCC has budgeted $11 million per year for eight years to develop CAD/CAM machinery to produce very large scale integrated chips.

4. Chip Packages - The housings for integrated circuits have between 10 and 40 connections for communicating with the outside; it is anticipated that VLSI chips will need perhaps as many as several thousand connections to input and output the volumes of data being processed.

The president of MCC has observed that each of these projects "is almost a limited R&D partnership".[108] Forming large limited R&D partnerships is still a new topic in the US semi-conductor industry, and many of the sponsors of SRC want to extend sponsorship beyond research to product development.

Also, SRC directors have been lobbying for using their North Carolina Research Triangle facility to focus on R&D for a 4 megabit memory chip. Most of the random access memory chips on the market today are 64K. Japanese chip makers have dominated this market and supply most of the world consumption, and in 1985 the new generation of RAMs -- 256K -- will come onto the market. It is expected that around 1987 or 1988 the capacity of a random access memory will have quadrupled to 1,000,000 bits and the next jump to 4,000,000 bits, is expected to occur in the early 1990s. However, SRC wants to leapfrog immediately ahead to the 4,000,000 bit chip and to achieve prototype production by 1987.

The SRC is operated by channelling research funds on VSLI design through the universities. These designs may be subsequently licenced back from academia, and this procedure allows SRC member firms to receive significant tax breaks for their investment while benefitting from the fruits of technology transfer.

US companies are being forced to team up because if they do not they will lose leads to foreign competitors. Also, IBM's investment in Rolm Corp. set the tone for other technology partnerships. For example, Trilogy Systems Corp. sold 15% equity positions to Sperry Corp. in mid-1983 for $42 million. Other reasons involve the fact that product life cycles in microelectronics are shortening, the rate of technological discovery is increasing and products have become more expensive. (Storage Technology Corp., for example, spent more than $12 million for its most recent generation of model 3350-type disc computer memory. By the time this firm begins selling the next generation of optical data storage technology, based on lasers instead of electronic signals for data storage, the firm will have invested $80 million.[109] To raise more than $40 million in capital this company formed a limited technology partnership for R&D with private US investors and also signed the DuPont Corp. as a technical partner in development of optical storage technology).

Another reason US firms are teaming up is that given the merging of computer and communication technologies for office and factory automation, the type of system functions that one must stay on top of are more complex than previously. For example, any new office automation system now must be able to handle images, voice, text and graphics, preferably simultaneously through windows.

Technical link-ups between firms may occur because of difficulty in rapidly learning telecommunications or data processing technology. For example, Northern Telecom Ltd. is setting up joint technology agreements with computer and software manufacturers.

A technical collaboration agreement is only as good as the paper it's written on, and collaborative efforts have become so important to the microelectronics industry that many companies have invested in partners to solidify the relationships. With the deregulation of computers and communications, many firms have realized that the traditional vertical integration approach to microelectronics may be self defeating. Indeed, within the past 18 months or so, Control Data has bought a 35% equity position in Centronics; IBM has purchased equity positions in both Intel and Rolm; Sperry has acquired a 15% equity position in Trilogy and a 13% equity position in Magnetic Peripherals. Similarly, Western Union acquired 25% of Vitalink and Control Data has bought into the Source Telecomputing.

From the perspective of the smaller companies, there are real advantages to such partnerships, such as expanded customer base, sales and company recognition. Few persons had ever heard of Vitalink Communications Corp. before Western Union purchased 25% of the firm in 1982 for $11.5 million. Another major benefit is a capital infusion. Limited partnerships with large firms is a real means of obtaining capital. The president of Trilogy, has remarked that:

> "It gets access to a much larger quantity of capital than just making the technology available through ordinary licensing." (110)

6.4.3 The Fifth Generation Response

The American approach to the fifth generation project is more entrepreneurial and diversified than Japan's highly coordinated research effort, described in the next section. As early as 1982 the Defense Science Board in the United States published a study which rank ordered technologies, including software technologies, of strategic interests in the US military. These are presented in Exhibit 6-30.

The United States Department of Defense's Advanced Research Projects Agency (DARPA) is putting a lot of money into universities and firms to fund AI software research. However, the initial focus of these projects is defense rather than commercial usage. One DARPA spokesperson has estimated that DARPA had spent around $50 million by the end of 1983 on AI software projects in businesses and universities.[111] DARPA, however, intends to eventually make this technology developed through its funding available to a wide variety of US industries. There are also private responses. ETA Systems Inc., a supercomputer spin-off of Control Data, is working on parallel processing computers and during 1983 IBM spent $3 billion on research and development in fifth generation processors.

The Los Alamos National Laboratory, the Lawrence Livermore Laboratory and SRI International have together formed a supercomputer research project (SPREAD) which is intending to develop a processor which can operate a thousand times faster than current processors.

EXHIBIT 6-30

US Defense Science Board List of Future Technology Priorities

1. VHSIC (Very High Speed Integrated Circuits)
2. Stealth bomber
3. Advanced Software (Artificial Intelligence)*
4. Computer-Aided Instruction (CAI)*
5. Fault-tolerant electronics
6. Rapid solidification technology
7. Machine Intelligence (Robotics/AI)*
8. Supercomputers (VLSI)*
9. Advanced composite materials
10. Monolithic focal-plane arrays
11. Radiation-hardened electronics
12. Space nuclear power
13. High-power microwave generators
14. Space structures
15. Optoelectronics
16. Space radar
17. Short-wavelength lasers

* These items represent areas with major AI components.

Source: Cited in Cognos.

Also, in late 1983 DARPA set up the Strategic Computing Plan -- focusing on AI software in defense areas such as expert systems, self-guided battle vehicles and rapid response battle management systems.

Rather than having a nationally coordinated plan like the Japanese, the Americans are clearly relying more on competition and market-determined forces. The rapid development of US-based AI companies and continuing extensive use of procurement policy by the US Department of Defense, it is hoped, will once again stimulate private sector innovation. The recent tax loosening of the US anti-trust laws as they pertain to cooperative R&D ventures and R&D tax incentives involving university contributions are also quickening capital flow.

Recent US tax changes have motivated private firms to channel research money into universities. Stanford University, together with 18 firms, set up the Centre for Integrated Systems in 1983 with initial funding of $19 million US, and several universities including MIT, Yale and Carnegie-Mellon, have AI labs which have been financed by the private sector. For example, DEC and IBM recently funded Project Athena, with a total funding of $70 million for computer-aided learning, and IBM, through the Semi-Conductor Research Corporation, has been financing VSLI research at a number of university research laboratories.

6.4.4 American vs Japanese Funding Styles for AI

It is instructive to compare the American and Japanese funding styles for artificial intelligence and software in general. The Japanese are, of course, involved in a large national effort with coordinated industrial/government planning. In the US, on the contrary, although a large amount of AI funding comes through military programs such as DARPA, the Air Force Office of Scientific Research, and so forth, these do not involve a nationally coordinated effort, and there are also major efforts, we have seen, in industry-led AI research -- such as the considerable support of university computer science departments by computer companies, such as Stanford's Semi-Conductor Research Association, MIT's Project Athena and Carnegie-Mellon's Robotics Institute. All of these projects have funding in the order of several dozen million dollars. We have also seen the initial formation of high tech cooperatives in the US to counter the Japanese effort, and there is the continuous infusion of research monies for smaller companies via venture capital funding -- a factor almost entirely missing in Canada -- where most computer research is supported by governmental funding agencies at significantly lower levels of support than in the US. One study has made the point that:

> "Even in the US, funding by traditional science and granting agencies would not be nearly adequate to support R&D into artificial intelligence and natural language processing." (112)

Although we are not reviewing British software policies as part of our country review, the British response to the fifth generation project must be noted.

The Alvey Report[113] recommended the launching of a national program for information technology, to be funded by 350 million pounds over half a decade. Recommending the doubling of British informatics activities, the report identified four critical technologies: software engineering and software tools; the human-machine interface (especially the development of speech based input and output devices); expert systems and intelligent data bases; and non-parallel processing computers, including VLSI chip research.

This collaborative effort among universities, industry and research organizations, will explicitly exclude foreign multinationals from participation, unless they can contribute a missing component of the program.

The British are developing what they call information systems factories, i.e. hardware/software systems for designing, specifying and prototyping software itself, and for computer-aided design tools in VLSI, all with networking capabilities.

Within the confines of the human/machine interface, the British research is concentrating on speech and picture processing and recognition, and within the area of knowledge-based computers (expert systems), the UK effort is addressing a wide range of research topics.

6.4.5 US Tax Treatment of Software

One of the main expenses for a software firm involves the development of new programs. The US has been trying to decide whether or not such outlays should be treated as research and development and given tax writeoffs or whether some or all of these costs might be "capitalized", i.e. recorded as an asset and written off slowly over several years.

According to a 1984 survey,[114] approximately 20% of firms in the US software industry capitalize some development expenses and consequently show greater earnings than if they had immediately written such costs off.

In an effort to prevent potential investors from being misled, the US Securities Exchange Commission on August 8, 1983, prohibited firms from capitalizing software development costs until this matter has been decided.

However, if capitalization becomes mandatory, as some have suggested, firms which have been writing off costs immediately would look as if they had much greater earnings.

With respect to software R&D exemptions, although it has been argued that virtually all software costs should be classified as R&D, and that there is no correlation between future software revenues and the amount of money spent developing a specific program, others have claimed that only the design and planning phase of software development should be given R&D writeoffs, and that costs such as those incurred in writing software codes should be treated as assets.

Before January 1983, IRS ruling 69-21 stated that software development costs could be written off as R&D expenditures (under Section 174 of the Internal Revenue Code) if: "the cost had been incurred in developing new or significantly improved programs or routines that cause computers to perform desired tasks."

However, in January 1983 in proposed regulation 1.174-2 the IRS stated that: "... the costs of developing computer software ... are not within the meaning of

Section 174." This proposed regulation also stipulated that some development costs of computer software may actually qualify under Section 174 deductions if and only if these costs have been incurred in the development of new or significantly improved programs or subroutines. Again, the import is to attempt to give tax breaks to projects which involve significant risk. However, the determination of "technical risk" is very difficult if not impossible for most software development activities.

By mid-1983 the Internal Revenue Service had withdrawn its proposed regulation and had agreed with the American Electronics Association, which held that appropriate regulations must concentrate on the end purpose of the software expenditure instead of the methodology of writing it, and that any costs involved in developing new or significantly improved products or production processes, irregardless of whether they are software or hardware, should be treated as R&D.

After the withdrawal, the US Treasury claimed, that the import of the proposed regulations under Section 174 had involved an attempt to give R&D tax incentives to "pure" software firms (i.e., firms selling software to end users and retailers while refusing these incentives to firms which merely develop software for internal use.

Presently in the US, software development costs of taxpayers in the "pure" software industry qualify under Section 174 for R&D write offs, but firms which produce software for internal use do not qualify. Furthermore, within "pure" software firms, mere modifications to existing software (such as transferring code from one type of machine to another) are not considered to be new products and do not qualify for scientific research tax incentives, while packages which add new functions to existing ones generally do qualify.

(US tax treatment of software and software copyright are further examined in chapter seven.)

6.5 The Japanese Model

According to the Japanese model, industrial development does not follow an R&D supply push but until recently has been based rather on licensing from external sources. Another aspect of this model is "controlled competition", according to which MITI equalizes access to foreign technology by the main Japanese firms to create internal competitive pressures and to also force the firms to further develop, modify and improve technology. Another aspect is industrial coordination of policies. MITI coordinates the support and introduction of new industries with policies to allow the sunset industries to die with a minimum of pain.

Most of the Asian Rim countries in fact have made extensive use of licensing of foreign technology, which ranged in 1979 from 0.2% of GNP in Japan to 0.05% in South Korea, and most of the Japanese computer industry's technology originated from a series of initial patents which IBM granted in 1962 to make mainframe computers in Japan, and from other patents granted by TI in 1968, which formed the technical base for integrated circuitry of NEC, Hitachi and Fujitsu.

On the other hand, the technology import/export ratio of many Asian countries has been increasing rapidly since 1978, and in 1975, for Japan only, increased 27%.[115] These trends are expected to continue in the next few years in spite of the significant expenditures on Japan's fifth generation project, which is really the first major original Japanese R&D effort. In the meantime, Japan continues to pay net royalties for technology to other countries of over $1 billion per year.

Under controlled competition, MITI ensures the the larger firms in any industry have equivalent access to new technologies. In the mid 1970s, for example, MITI temporarily postponed an agreement between Hitachi and RCA for integrated circuits until a main competitor of Hitachi's found an equivalent licensor. (Thus it is surprising that the Japanese have recently agreed to share their technology with the Koreans through a "window" organization, which will "monitor Korean demands for more access to Japanese technology".)[116] In 1982 alone Japan paid in excess of $1.8 billion in licensing fees for imported R&D, approximately half from US firms, and many American firms are resentful that this licenced technology served as a base for what became a major industrial competitor. Although the Japanese clearly wish to avoid a similar situation with Korea, given Japan's trade surpluses with a number of Asean countries, it will have to make major concessions in technology access. However, the Japanese are tying these arrangements to Japanese exports.

6.5.1 The Japanese Computer and Software Industries

The Japanese computer market is the second largest in the world after the United States, and Japan exported $2.7 billion worth of computers in 1983.[117] IBM's CEO has estimated that Japan's combined data processing and telecommunications industries comprised more than $30 billion in 1981 and will grow to twice this figure by 1986. Although the Japanese are not presently strong global competitors in the software field, and also had less than 2% of the US market for all types of computers in January 1984, within the decade they will be a major threat and IBM is anticipating that within a half decade its real rivals will no longer be Honeywell, but will be Japanese firms which are totally vertically integrated from chips to mainframes and which will have software divisions for each new product line which they enter.

Most of the Japanese software houses have less than 80 employees, and the largest independent suppliers average 150. There were 1,965 Japanese companies supplying software in January 1984. Less than 50% of these had half the total revenues, and only seven firms have revenues greater than $25 million annually. These 1,965 firms collectively employed 153,000 persons in mid-1984.

Since applications are mainly written by users, packaged products have been slow to develop, rising from $0 sales in 1977 to less than 10% of the total domestic market in 1983. Of the total software workforce, less than 25,000 persons are working on packages.

Ironically, IBM's domination of the American perscom market, with such developments as the PC, could actually increase the Japanese market share of the American perscom market. The brokerage firm of Jardine Fleming in Tokyo has reasoned that increasingly, software writers in the US will write for IBM machines and the Japanese, with their brilliance at mass production, will then be able to turn out extremely cheap competing models that run these same programs.[118]

The current battle in the Japanese computer market between IBM and her adversaries (Fujitsu, with 1983 sales of almost $4 billion; NEC Corp., with 1983 sales of $6.1 billion; Hitachi, with 1983 sales of $16.3 billion) is centering around computer software and personal computers for office use. The software battles are described in the section on software copyright, but the significance of the personal computer in offices is extremely important, due to the intricate form of the Japanese written language. Using conventional typewriters to print the literally thousands of characters in Japanese is impossible. However, with a perscom plus printer there is little effort involved. Typically, without a personal computer a Japanese office worker will communicate through slow, handwritten memos, and there is a huge market for word processing software and hardware which can handle iconic forms of Asian languages in countries such as Singapore, Taiwan, China and South Korea.

Also, as in the United States, Japan is currently allowing the merging of computing and telecommunications companies, hitherto forbidden. For example, Nippon Telephone and Telegraph Co. is creating a large integrated network, spending $7 billion per year over the next few years to create a net of paired wires, fibre optics, etc., which can handle digital, voice, videotex and computer data. (To get work on this network, IBM Japan Inc. has formed a partnership with Mitsubishi and a Japanese software company.)[119]

Although the Japanese have automated more factories than any country, many Japanese offices resemble those of the nineteenth century, with telephones being used mainly for personnel messages, with idiosyncratic filing systems and hand-written memos. Japanese documents are usually written by hand and not on a typewriter keyboard, because of the iconic character of the Japanese language. Naturally enough, office productivity is even lower than that of the US and European countries, and there is presently great enthusiasm for office automation. Recent technological changes such as Japanese word processing have made it possible to cheaply store and process virtually thousands of kanji symbols, which comprise written Japanese. Although the Japanese Office Industry Association states that integrated office equipment (business minicomputers, perscoms, facsimile, Japanese word processors) totalled only $2.2 billion in 1981, this figure will exceed $6 billion by 1985.[120]

Given the imminent incorporation of expert systems into office automation, it is important to note that there systems are "language specific". Because analyzing a natural language depends on its syntax, an expert system cannot be readily converted to perform analysis in some other language. Software which utilizes such algorothms is almost impossible to translate for use in a different country. This fact may mean that expert systems written in Japanese are not going to be particularly exportable to North America, and vice-versa.

6.5.2 The Japanese Software Effort

It is often claimed in western countries that in areas of original research such as software production, which often resembles the chaotic process of artistic creation, the Japanese, with their structured Shintoistic traditions, are not very good and have succeeded thus far mainly by licensing and modifying technology from the west (more than $10 billion worth in the last decade).

This belief may not prove well founded, and in spite of the fact that in recent lawsuits a number of Japanese firms have been publicly degraded by being forced to open their products for inspection by IBM to make sure that they are not copying IBM software, the Japanese are now putting major efforts into the development of software.

Many of the big hardware manufacturers such as Fujitsu, Hitachi, Mitsubishi and NEC have thus far put comparatively little effort into software development. However, given the fall in comparative price of hardware compared to software, all these firms have announced a shifting of research resources into software development. Exemplary of these developments is Hitachi, which in the past three years has increased their R&D software budget from 10% of total R&D to just under one-third. Similarly, Fujitsu now spends one-third of its total R&D budget on software, but this is written mainly for Fujitsu's mainframe computers and sold domestically in Japan.

In 1982 IBM sued Hitachi for allegedly copying and modifying IBM software and reselling it in embedded form in Hitachi computers. In an out-of-court settlement, Hitachi agreed to pay IBM $4 million per month for software licensing fees and also is allowing IBM to inspect any new Hitachi products before they enter the market to make sure they do not infringe on any IBM software copyrights. Fujitsu similarly has agreed to an out-of-court settlement to pay IBM an undisclosed amount, rumoured in the millions, and also has agreed to not copy IBM's software.

The Japanese have not had significant software exports outside of Asian countries which have iconic languages. Due to a number of sematic shifts and other problems, when programs are translated from English to Japanese or vice-versa, they often have to be totally rewritten, and business accounting practices may be totally different cross-culturally. For example, Japanese and British accounting practices are so different that Japanese accounting programs are not useful in the US or Britain.

The Japanese have also been comparatively slow to develop off-the-shelf software packages and have usually produced customized software. For all of these reasons, an independent Japanese software industry is developing only slowly.

But in the area of embedded software, for example in video games, cultural differences do not count, and Japan has already captured a significant part of the US market for this type of software. In other areas, such as software for scientific and engineering purposes, procedures have been standardized all over the world, and cross-cultural differences are insignificant inhibitions to export.

6.5.3 Software Factories

One of the basic forms of producing software in Japan involves the software factory. Toshiba, for example, has a software factory of 2,800 software developers focusing on industrial CAD/CAM software and plans to increase this number in the near future by 2,000 additional programmers. The NEC Corp. is also spending up to one-third of its $400 million research budget for software in 1984 on productivity aids to improve the quality and amount of Japanese software. NEC has also started its own series of software factories. The utility of such software factories, however, remains to be proven, and the process of writing software is still a labour intensive, sometimes almost anarchistic process, with development time still quite unpredictable. In fact, we saw software writing does not follow normal rules which govern industrial processes. For example, when one adds more people to a software project, the development time is sometimes likely to be increased.

6.5.4 Government Support for Software

In 1981 the Information Technology Promotion Agency of Japan's Ministry of International Trade and Industry set up a software technology centre which collectively involved software writers from the universities, industry and government. The Japanese private sector is also continuing the practice of starting new software subsidiaries for each new product area that a firm enters.

Most significantly, the government allows 50% of revenues from sales of packaged software to be tax-exempt for four years. But the firms must re-invest these monies in further package development. MITI also has put up significant amounts of monies in loan guarantees for software development, since the Japanese banks are as stogy as ours with respect to software funding.

MITI is also making a major thrust into the export market with the January 1983 signing of a treaty for China-Japan technical cooperation in software. Fujitsu and NEC have opened training centres in China, and the Japanese clearly hope to replicate their success in Singapore.

Of special interest is Japan's fifth generation computer project which is focusing on parallel processing computers and AI software. Begun in mid-1981, the government is supporting 35% of the development costs of this project and has invited firms from all over the world to join in. In 1984 the software component of this project is budgeted for $23 million, and MITI has also set up tax breaks and low interest loans to software producers.

As discussed in the section on copyright, MITI is also attempting to nurture its software industry with proposed revisions in the copyright law by basically making it legal for Japanese firms to copy existing software without the permission of the owners.

In the meantime, US software producers are still significantly ahead of Japan. In 1983, for example, the total Japanese software sales were still between one-quarter and one-third of those of the US firms.

6.5.5 MXS

The much vaunted Japanese perscom invasion of 1982 was a failure because of the lack of software for the machines and quite eccentric documentation. The Japanese actually first exported perscoms to the US in 1980, the year before IBM's PC, but today IBM still has 30% of the US market and the Japanese only 2%. However, as The Economist has noted:

> "If history is a guide, Japan's near absence from the higher (personal computer) end of the microcomputer market should worry the Americans rather than relieve them." (121)

If Japan enters the cheap, low end of the perscom market, as they did with color TVs, VCRs and pocket calculators, it is once again Japan's consumer electronics producers rather than big computer manufacturers that will open up this market, which in 1988 will be worth $40 billion world wide.

Recently, in fact, 14 Japanese firms signed an agreement with Microsoft (the American software company, whose operating system MS/DOS is used in IBM computers) to use MSX/DOS in their perscoms -- a clone and improvement of MS/DOS. The MSX operating system works on 8 bit perscoms, but most American machines are already 16 bit, and 32 bit perscoms and soon expected to come onto the market. The Japanese, in other words, are once again apparently going for the cheap end of the market.

MSX was conceived by Matsushita Electric, the biggest producer of consumer electronics in the world, and developed with the Askii Corporation of Japan. The Japanese firms of JVC, Sharp, Sanyo and others are planning to mass produce and export an MSX machine. A number of marketing firms believe that the Japanese will have this machine in US department stores by Christmas 1984 and that it will sell for between $200 and $300 US.

American software firms are also writing software for these Japanese machines, something that no one ever believed they would do, and the Japanese have been recruiting software documentation writers from the Silicon Valley.

One of the main reasons that IBM purchased a 12% share of Intel in 1983 and essentially abandonned vertical integration was to block the Japanese in replicating with computers what they have done with colour TVs -- namely "low bidding and undermining the infrastructure of the component suppliers that supported the industry".[122] In other words, innovation isn't paying for itself because the Japanese have upset the economies of the industry permanently, and within several months after a new chip is placed onto the market, the Japanese have now produced a competitive chip with vastly reduced price. As a result, US chip makers can't recuperate their investments.

6.5.6 Japanese and American Chip Competition

Due to a number of reasons, by late 1979 the Japanese chip producers had made their first major coup and had won 42% of the US market, for the then contemporary generation of memory chips, the 16K RAM. In the next generation of chips, the 64K RAM, the Japanese again beat the US producers. In 1982 Japanese chip makers captured 70% of the world market for 64K RAMs. It is also thought that the Japanese will capture half of the world market for 256K random access memories, but they are also diversifying into other chip markets beyond memories and hope to become major suppliers of semiconductors.

NEC and Fujitsu for example, are both concentrating now on logic chips, and all of Japan's 13 major chip producers have heavily entered the gate array market. The importance of gate arrays has already been discussed and although these customized chips comprised world sales of only $150 million in 1982, by 1985 it is expected that this market will be over $500 million.

It is instructive to compare spending on new information equipment for the US and Japan. With respect to the main producers of integrated circuits, spending is rapidly becoming equal. However, Japan has been spending a larger percentage of sales on expansion of capacity than has the US. In 1981, for example, Japan's capital investments were $838 million, 25.4% of sales, an increase of 27% over 1981. However, the nine largest US manufacturers in 1982 made a total capital investment of $1.18 billion, an 11% decrease from 1981.[123]

In fact, it is thought that in 1982 the total number of Japanese produced semiconductor devices already exceeded that of American firms for the first time, and the US was first in dollar sales only because it produced more sophisticated products.

Also, in 1982 Japanese producers doubled their exports of integrated circuits to the US from $252 million in 1981 to $532 million.[124] In contrast, US exports of integrated circuits to Japan increased to only $159 million, up from $133 million in 1981.

In summary, although until recently the US had led Japan in production of total number of integrated circuits ($2.4 billion as opposed to $1.1 billion), Japan's IC production volume now exceeds that of the United States. (Canadian production of integrated circuits is miniscule, lower than West Germany, the UK, Italy and the Netherlands and exceeds only Belgium).

Control of chip markets give certain software advantages, and it is widely expected that the Japanese will solve many of their present software development problems within the decade and that Japan will increase its current tiny share of the system's software packages, either locally developed or copied from the West, to around 6%-10% of world markets by 1988. Ulric Wile has remarked that it is only after the Japanese can really compete in software that world software prices will radically drop. However, most standard programming languages such as Fortran, PL/1, Basic and Cobal have English prerequisites, and it is also thought that the Japanese will continue to have a rough time with the conceptual part of software development.

In the meantime, the Japanese are sharpening their competitive hardware advantage with major advances in the use of semi-intelligent robots for flexible manufacturing and factory automation in general.

It is also thought that the Japanese penchant for aggressive international trading practices -- including export support tactics such as indirect import restrictions, selective quotas, and predatory pricing -- will continue and be extended to software areas. Anticipating US and Western European retaliation against present export/import policies, many Japanese high technology firms have shifted manufacturing facilities to the US and Europe. As a result the data processing industry is becoming quickly internationalized and in the first two quarters of 1984 more than 15 major US-Japanese partnerships were set up.

The Japanese meanwhile continue to dominate in many areas of computer hardware beyond manufacturing systems, including facsimile machines, hand-held computers and 16K semi-conductor devices such as 16K and 64K RAMs. They have also entered smaller markets such as those for PABXs, IBM plug-compatible computers, dot matrix printers and copiers, and have plans for export of semi-intelligent adaptive (learning) robots, personal computers based on 32 bit microprocessors and supercomputers. In addition, a small number of Japanese firms have entered the markets for laser printers and intelligent copiers.

The Japanese are still behind the US computer industry in hardware and software for office automation. Both Hitachi and Fujitsu have a strategy based upon IBM software compatibility, but both firms' overseas marketing has been slow, and there are real problems with pursuing a plug-compatible strategy with IBM, since IBM has shortened its products' life cycles, and plug-compatible manufacturers no longer have a sufficient amount of time to decipher new IBM designs. This shortened product life cycle eliminates many advantages of reverse engineering.

Presently then, Japan, like Canada, has virtually no significant share of offshore systems software or applications software markets. The Japanese, well aware that software is becoming an increasingly larger value-added component, continue to devote high outlays to R&D spending, focusing this effort in the area of applications software. Still they have not come up with any useful methodologies for the much vaunted automated programming, and are still having great difficulty due to their languages and other documentation problems with serving software end user needs overseas.

Since almost all software documentation is written in English, Japanese programmers must first learn this language, and their programmers are no better at learning natural languages than ours are. Secondly, the design of systems software is still essentially an art rather than a disciplined procedure and requires a unique blend of conceptual and intuitive thinking, while the Japanese are still more proficient at developing hardware. For all of these reasons, over the next half decade the Japanese may not produce software for the export market that is intrinsically better or cheaper than rivals in those countries.

More significant is the fact that the Japanese will be copying and modifying North American software packages for resale at costs significantly below those of the original, as is now the case in Singapore and several other countires. Legal protection in selling such copies thus becomes important.

6.5.7 The Fifth Generation Project

Japan's fifth generation computer project is the first to attempt to methodically develop parallel processing machines which can simulate thinking, and generalizing, input symbols such as visual or spoken patterns, words, entire sentences, etc. It is expected that fifth generation computers will be able to do most jobs which mid-level managers do today.

The transition between Von-Newmann and parallel processing computers will come to be seen as a watershed in computer developments, and several countries have seen this as a period of technological change in which they may whittle down the US's dominance in the software field. For example, several European countries and several hundred western European companies have banded together in a five year $1.1 billion program called the European Strategic Program for Research in Information Technologies, and we have seen a similar collaborative effort in Britain between industry, government and universities.

These projects are concentrating on advanced computer technologies such as parallel processing hardware and AI software. As of this writing, however, no real parallel processing computers are on the market, but every major computer manufacturer in the US is engaging in parallel processing research.

The Fifth Generation Project is being run under Japan's Institute for New Generation Computer Technology, formed in mid-1982, funded to date at $51.3 million (Canadian); over the next decade it will receive over $500 million and is expected to involve several hundred researchers. The Japanese are attempting to make AI-based computers and software which, when combined, will be able to store images and recall them by associative patterns, a system which can at least simulate cognitive and associative human reasoning, which can learn and program itself like the human brain, which can contextualize and infer meanings and input and output data in a variety of visual and verbal forms. MITI has also set itself the explicit goal of capturing almost one-third of the world computer market by 1990.

At the heart of the fifth generation project is software research on natural language processing, because non-programming users will be the target group; research in the fifth generation project includes analysis of speech waves, analysis of syntax, and phonetic and sematic analysis.

Due to the their printed language, the Japanese are quite naturally concentrating on machine translation and have set themselves the formidable goal of 90% accuracy in machine translation, with the rest being translated by humans.

The critical part of the fifth generation plan involves the development of "knowledge based inference systems" or expert systems. Since one of the goals is to make computers widely accessible, the Japanese have realized that inputs should be in natural languages, and for the computer to be able to contextualize (e.g. distinguish spirits as "wine" from spirits as "ghosts"), it must have a great deal of knowledge about the domain of discourse and about the user. A computer in other words, must first be given basic facts concerning both the domain of the users' knowledge and the inference rules he or she uses to think about this domain to itself make inferences which are based on the knowledge.

The core of the fifth generation project is software, focused into three areas: problem solving/input software; knowledge-based software; and software for intelligent interfaces.

Problem solving/input software is involving research in developing a new coding language for problem solving, which can accommodate object oriented programming and logic programming; research on algorithms for problem solving; and the development of an actual inference machine which can perform 100-1,000 million inferences per second.

The second component, knowledge-based software, involves R&D into AI-based techniques for ordering and storing human knowledge. Research here is being conducted on means of acquiring, representing and learning knowledge. One of the explicit goals of the knowledge-based computer is to perform inferences from 100 million items of data and 20,000 inference rules.

The third component, software interfaces, is concentrating on speech communication with computers in a natural language. The speech recognition goals are startling -- that the final system be able to identify speech signals as rapidly as a human can, i.e. almost immediately; that it speak in both Japanese and English, and that it rapidly adapt itself it to unknown speakers.

The fifth generation project is being coordinated by the Institute for New Generation Computer Technology, and all hardware and much of the software is being contracted out to Japanese firms. Member organizations of this project are shown in Exhibit 6-31. Composed of an industrial/government mixing of personnel, researchers from industry are rotated every three years back to their firm to encourage knowledge transfer to member firms.

The initial knowledge-based computer in this project is intended to have a microprocessor architecture which is similar to those of conventional computers rather than the more advanced parallel processors which are planned in subsequent steps.

The Japanese have set themselves a number of formidable specific goals:

1. The targets of the machine translation project involve a system which can handle machine translation at 90% accuracy with a range of 100,000 words and produce Japanese/English translation at a cost of less than 30% that of human translators.

2. Another objective is to develop knowledge-based computers with specialized uses such as computer-aided design and robotics.

3. The development of a speech understanding system, including phoenetic typewriter and speaker identification.

4. The development of an image/picture data and retrieval system containing 100,000 retrievable image frames, with a retrieval time of 100m/sec -- within this area the Japanese are blending language and image retrieval so that images may be retrieved via a spoken command.

5. The development of a knowledge-based problem-solving system which hopefully will be able to combine, test and discover mathematical algorithms.

EXHIBIT 6-31

Participants in the Japanese Fifth Generation Computer Project

<u>Industrial Participants</u>

Electrotechnical Laboratory
Fujitsu Laboratory
Hitachi Ltd.
Japan Electronic Computer Co. Ltd.
Matsushita Communication Industry Co. Ltd.
Mitsubishi Electric Co. Ltd.
Nippon Electric Co. Ltd.
Nippon Telegraph and Telephone Public Corp.
Oki Electric Industry Co. Ltd.
Toshiba Corp.

<u>University Participants</u>

Keio University
Kyoto University
Seikei University
Tohoku University
Tokyo Institute of Technology
University of Tokyo
Waseda University

Source: Cognos

We have now reviewed the main software policy measures and development goals of four other countries: Brazil's protective, development-oriented stance, Singapore's export-oriented position, the entrepreneurial approach of the US and Japan's fifth generation effort. Although these policies are too recent to have been evaluated, it is clear for reasons subsequently described that protective measures (e.g. a software import ban or software tariffs) would be inappropriate for Canada. However, equally clear is the fact that elements of each of these policy orientations are relevant to the Canadian situation; e.g. MITI's coordinated phasing in of high growth industries with gradual phasing out of sunset industries. Given finite resources, Canadian policy makers cannot appropriate necessary funds to support new high growth industries such as software, while at the same time continuing to pour hundreds of millions of dollars into dying industries to save jobs in the short term. Policymakers cannot continue to dilute resources to impotent dabs, as occurred when a "microelectronics development centre" was placed by the federal government in virtually every province. Canadian policymakers, in the name of regional disparity, can no longer place productivity centres into cities which handle less than 3% of Canada's business.

With limited resources, a nascent software industry which has existed only a few years, and cutthroat world competition, it is crucial that we apply policy instruments which are competitive with those of other countries. These instruments are the subject of the following chapter.

7.0 CANADIAN SOFTWARE POLICIES

7.1 Introduction

In this section we are concerned with evaluating the tools and instruments available to promote the growth and economic viability of the Canadian software industry. The Department of Supply and Services has recently announced that it is about to release new guidelines on software procurement. Increased government procurement is regularly called for in study after study, but government procurement of goods and services makes sense, and has proven to be a useful development instrument <u>only when there is a real user need</u>. Otherwise, procurement may involve significant market distortions -- the government purchasing technologies and services that no one else would buy, as happened with Telidon hardware, and entire industries being dependent on a single client, the federal government.

The economic basis for government intervention into industrial processes ultimately lies in the existence of "market failures" which occur when one or more conditions are present that preclude the efficient allocation of resources by private transactions. When market failures for a set of economic activities are insignificant, a strong presumption exists that government policy will not improve allocative efficiency, although it might help realize other public policy objectives.

Although the objectives of the federal government vis-a-vis the software industry are not entirely clear, the main objective is taken to be the emergence of a world-competitive, domestic software industry.

As a starting point, we need to consider those attributes of the software industry in Canada that suggest a need for government intervention (of some sort) to augment the workings of the marketplace. Much of this discussion has been foreshadowed in earlier sections. Nevertheless, it is useful to bring the preceding discussion together in an integrated fashion. Exhibit 7-1 summarizes the main strengths and weaknesses of the Canadian software industry, and Exhibit 7-2 presents software opportunities for Canada in two regions of the world: North America and in Pacific Rim countries, such as China, Singapore, Malaysia, South Korea, Indonesia

EXHIBIT 7-1

Strengths and Weaknesses of the Canadian Software Industry

Strengths

A developing packaged business sector; a nascent AI industry; expertise in custom software in resources, energy, telecommunications, finance and training and education.

Proximity to the largest English speaking market in the world for software, the US.

Proximity to the Asean region, the area of the world with highest consistent growth rates.

On a world scale, several first rate universities with expertise in computer science.

A stable economy and political system.

Weaknesses

Small domestic market, with 68% of Ontario's software market the domain of US subsidiaries.

Highly fragmented in structure; a few big companies but mainly small ones.

Absence of sufficient capitalization; with many companies experiencing severe and constant cash-flow problems.

Lack of marketing expertise; weak venture capital markets; conservative banking industry.

Absence of federal support in a recessionary period; imposing of conflicting tax rulings on software; absence of tax/fiscal and R&D incentives given to traditional manufacturers and software producers in other countries.

EXHIBIT 7-2

Software Opportunities for Canada

<u>Canada and the US</u>

Traditional business packages for micros (i.e. billing, inventory control, accounts payable, accounts receivable, spreadsheets, general ledger, payroll, management support, etc.).

AI-based business applications for 32 bit micros (database management, query languages in French, intelligent spread sheets, sophisticated graphics, etc.).

Transaction processing software for use with commercial applications such as cheque processing, order entry, transactional Videotex-based banking and shopping, tiered cable TV services, etc.).

Telecommunications, networking software (common carriers, cable, satellite, HDTV, mobile communications, etc.).

Programming/tools applications generators.

Educational/training software (content, authoring tools, authoring systems, AI-based tutorials, expert systems, etc.).

Industry specific software (generalized programs which can be utilized by many firms within the same industry, such as forestry, mining, petroleum exploration and processing, agriculture, finance and health care).

<u>Pacific Rim</u>

Applications business and management software for micros, minis and mainframes (financial planning, cash flow management).

Applications generators in a wide variety of industries.

Transaction processing (especially in countries such as Singapore and South Korea).

Customized software in large telecommunications, resource and energy development projects for mainframes and minis.

Customized and packaged software for education, for internal training in large corporations (the possibility of using Ontario's educational micro as a vehicle for Canadian software); English-training, etc.

Machine translation and natural language processing software for governments, large firms, etc. (English is the business language of several Asian countries).

Expert systems in resource exploration/ satellite sensing.

Manufacturing resource planning (from production planning to inventory control to materials planning).

EXHIBIT 7-2

(continued)

Canada and the US	Pacific Rim
Office and factory automation software (distributed processing, networking, LAN/compatability software).	
Standard software products for mainframes -- in spite of the increased number of micros, the mainframe program market is still growing at about 28% yearly. Maintaining and modernizing old programs is still a significant portion of the North American and European software markets.	

and in China. The Asian countries of the Pacific Rim are of special interest, given the extensive development projects (described in the section on exports) and the fact that in the last couple of years, Canadian trade with these countries exceeded trade with Europe for the first time. Estimates of opportunity areas have been based on:

1. the industrial, marketing and business trends delineated in this work
2. more than fifty interviews with government/industry representatives from the respective countries concerning present and future software needs.

Since recent software studies have repeatedly documented potential European markets for differing types of software, in this Exhibit we are emphasizing the US and Pacific Rim opportunities.

7.1.1 Potential Sources of Market Failure in the Software Industry

Inappropriabilities: Market failures often are associated with activities where it is difficult or impossible to charge users of specific services directly according to use. For example, direct charging may be impossible because non-payers cannot be excluded from utilizing the service once the service has been provided for any one payer. In this case, a smaller amount of output will be produced than would be the case in the absence of "free-riding" by consumers or other producers.

The presumption that it is difficult for firms to appropriate all the benefits of the research and development (R&D) they undertake underlies government support for domestic R&D expenditures. For analogous reasons, it might be argued that software firms cannot appropriate all the benefits of their design and development activities. Rather, some of the benefits may be appropriated, in part, by software users or by other producers. Most directly, there is the potential for "piracy", whereby users copy software. In other cases, there is the potential for certain software concepts to become industry standards, e.g. the spreadsheet concept in Visicalc. Given the ability of other firms to duplicate a successful software concept, it is unlikely that software innovators will be able to fully capture the returns associated with a new product.

While it is certainly true that piracy and other appropriability problems impact the Canadian software industry, it would not seem that these are major sources of market failure. The piracy problem tends to be relevant where the user requires minimal guidance in using software, as in the case of straightforward business applications software such as spreadsheets. By contrast, it is much less feasible for users of expert systems to simply "copy" programs without the support services of the software producer. In other cases, e.g. systems software, the relevant programs are being increasingly tied into hardware, i.e. firmware, which makes it easier to protect the benefits of a program, since the price of the embedded software is included in the hardware purchase. With respect to applications software, there is evidence that the "life-cycle" of products is becoming shorter. If true, it suggests that the benefits from appropriating features of rival software products may be limited, since a product may be approaching market maturity by the time its features are widely copied.

Inadequate Information: Market failure can also occur if individuals are not sufficiently well informed to permit them to make rational decisions in the light of their preferences. In a world of increasing specialization and complex technologies, it may not be possible to extract from observed characteristics the information required to undertake relatively well-informed transactions for a wide variety of goods and services.

Information available to the market can be inadequate for several reasons. One is the condition that the information is usually costly to produce but can be disseminated at relatively low additional cost. Social efficiency would require that the charge for information not exceed the marginal costs of its distribution. If this were done, however, the supplier of the information might not be able to recover his total costs of generating it. The result might be a less than optimal amount of information being produced and disseminated. A closely related notion is that it may be difficult for the producer of information to retain proprietary ownership if the information were widely disseminated through private exchange. In this case, production of such information might be seriously retarded. These market failure properties of information distribution are an ostensible rationale for government actions such as R&D grants and patent protection.

Information inadequacies are relevant in the economics of the software industry. In particular, failures may occur in financial markets if risk-averse potential investors require substantial amounts of information about the specific software development plans of producers, while the revelation of this information is either extremely costly to generate, ex ante, or would threaten proprietary ownership of the software. In effect, such information inadequacies would place a premium on firms being able to finance software development through internally generated funds.

The efficiency of financial markets is potentially damaged by a number of factors. One is related to provisions that enable business losses to be carried over from one taxation year to the next. The losses for a given business year may be carried back to the preceding year and forward to the following five years. For a new company, tax reductions for losses are automatically deferred; for a company that takes more than five years to become profitable, they may even be lost totally.[125] The tax loss feature is inefficient and possibly unfair, in that government sharing of profits is immediate wheras sharing in losses may be deferred. In addition, if the loss is carried forward to later years, the firm's cash flow will increase only when the firm becomes profitable again, i.e. when the tax credit has lost part of its usefulness as a source of financing.[126]

The foregoing feature of the tax system has been recognized in respect of R&D activities, and policies have been put in place to deal with the consequent capital market biases. One such policy is the recently implemented scientific research tax credit. (In a later section we evaluate this instrument.)

Capital market imperfections may also be related to restrictions on eligible investments by financial institutions. Specifically, performance clauses pertaining to the stability of earnings and dividends may limit the access of smaller firms to external equity capital. The preference of Canada's major financial institutions for investing in relatively large, dividend-paying companies leaves smaller software companies to obtain equity financing largely in the venture capital market. For reasons we will discuss below related to economies of scale in risk diversification, it may be unrealistic to expect Canadian venture capital firms to fill the equity financing gap created by the investment preferences of established financial institutions. (One point that might be noted in this regard is the relatively high cost of new share issues.) The Economic Council of Canada estimates that the underwriting process requires owners of small and medium-sized firms to sacrifice about 50% of the value of their shares to gain access to the public equity market.[127] An implication of these observations regarding inefficiencies in Canada's financial markets is the potential need of policies that promote internal investment funds in emerging software companies.

Information failures may characterize other activities, besides R&D, where the production of information is costly or while the dissemination of information is relatively cheap. In particular, market research and intelligence activities may be subject to marked economies of scale, although other activities in the production process may favor smaller-sized units. What would seem to be required in this case is the emergence of specialized market intelligence companies who sell information -- on a pooled basis -- to smaller producing companies in the industry. Where these specialized companies can retain some proprietary control over the common information base, the market solution may work tolerably well, especially when the specialist information gatherer can impose two-part pricing schemes, e.g. a retainer fee plus a small charge per report. This latter approach is most likely to be viable in industries where there are a relatively small number of large companies who can justify a retainer relationship. On the other hand, where the potential users of the information are small and fragmented, two-part pricing schemes are unlikely to be viable. But charging the full costs of producing the information -- rather than the relatively low marginal costs of distribution -- will lead to an inefficient restriction in the amount of information disseminated. A possible solution is to have some government agency disseminate the information at marginal cost.

The relevance of this latter potential market failure feature of the software industry is highly speculative. There are clearly private companies in Canada that specialize in market intelligence for software producers. Whether there is an "undersupply" of market intelligence is another question. At least in principle, however, there is a rationale for specialized government information services regarding marketing opportunities for domestic software producers.

Economies of Scale: Another efficiency-based rationale for government intervention arises from the existence of great economies of scale, i.e. decreasing unit costs over substantial output rates. The existence of significant economies of scale introduces a potential conflict between efficient production and a larger number of competitive suppliers. This is an especially relevant concern when the sustainability of the efficient production structure is threatened. That is, where existing prices do not discourage inefficient entry. The theoretical underpinnings of this concern have been developed in the context of the debate surrounding deregulation of the telecommunications industry.[128]

Another aspect of economies of scale that suggest a potential role for government intervention are economies of risk spreading. It is well known that the variance of returns is reduced by pooling projects whose returns are imperfectly correlated. A potential argument for government intervention exists in this context if there are reasons to believe that the private capital markets cannot pool risk efficiently.

In an earlier section, we discussed several possible sources of inefficiency in Canadian venture capital markets. They related to biases in the tax code, regulatory restrictions on eligible investments by financial institutions, and barriers to competition in the financial sector. We acknowledged that these sources of inefficiency constituted a potential rationale for public policy. Additional motivation for direct government intervention would exist if government agencies had lower transactions costs than private sector investors in pooling risky investments, and/or if potential problems of moral hazard were less severe in public investment agencies.

Transaction costs refer to the costs of identifying and making investments in any given set of projects. Moral hazard refers to the possibility that self-insurance (through portfolio diversification) will lead an investor to be less careful (on the margin) about the quality of his investments, or to formulate less precise estimates of the cash flow characteristics of given projects. Where higher transactions costs and greater moral hazard characterize private sector venture capital investing, there is an argument, other things constant, for venture capital investing by government agencies. But, in this instance, there is no persuasive reason to believe that, on either dimension, portfolios of software investments held by public agencies will be more efficient than those that would be created by private agents.

In short, we do not believe that market failure considerations related to natural monopoly attributes have a compelling relevance for the software industry.

Merit Goods: One additional argument that is often made for government intervention is that an activity is intrinsically beneficial for society, although -- for one reason or another -- the benefits may not be recognized or widely appreciated. This argument has been made in the context of policies to promote Canadian culture, and to reduce technology imports from outside Canada. It has also been invoked in arguments about the quality of jobs in "high technology" sectors compared to jobs in other sectors of the economy. Without necessarily dismissing the general argument, we cannot see any special applicability of a merit goods argument for government intervention in the software industry. For example, much of the employment created is of a fairly routine nature and is increasingly susceptible to automation.

7.1.2 Other Motivations for Public Policy

A number of specific objectives recommended by some observers for public policy in this sector do not fall cleanly into one or another of the market failure categories described above. For example, a recent government-funded report identified management inadequacies in marketing expertise and strategic planning as major barriers to the viability of British Columbia's fledgling software industry.[129] The study's authors were not able to identify many potential government responses to this problem, other than perhaps increasing financial assistance for software development, increasing purchases of BC software by the provincial government, and undertaking more trade mission activities.

Our own interviews with software developers, as well as our review of the relevant industry literature, confirms the importance of marketing expertise in the applications software market. However, it does not necessarily follow that government intervention is either warranted or desirable. In accord with our preceding discussion, a significant source of market failure must be found to exist before plausible efficiency-based arguments for government intervention can be made.

One potential argument is that there are appropriability problems with respect to marketing expertise, such that successful marketing strategies are easily appropriated by rivals, thereby depriving innovators of an acceptable risk-adjusted rate-of-return. If this is a significant phenomenon, it could contribute to a scarcity of marketing expertise, since returns to the marketing function might be uncompensatory. While the argument cannot be dismissed out-of-hand, it seems fairly weak. For one thing, much of the customer good-will created by effective packaging, displays, advertising, demonstrations and the like will be impounded in the company's trade name or product brand names, thereby providing the company protection against excessive "free-riding" by competitors. Support for this assertion is IBM's major and long-standing competitive advantage already derived from the company's obsession with providing high levels of customer service. For another, an important marketing edge -- especially in packaged software -- is manuals.[130] But manuals themselves are subject to protection under the Copyright Act.

Another argument is that the industry itself cannot train sufficient personnel to undertaken critical functions, including marketing, and that education and training are the responsibility of the state in any case. Note that this assessment argues for policies to promote the supply of marketing expertise, rather than the demand for marketing expertise, which would be the impact of government procurement policies. The issue of to what extent technical training should be a responsibility of government rather than of industry is a complicated one, and it is well beyond the scope of this study to examine. However, we would note that, in most industries, training provided in the public education system must usually be substantially augmented by on-the-job training and experience.[131] Hence, the ultimate requirement for the companies themselves to invest in company-specific marketing and other technical skills seems inescapable. The companies have argued that they do not have the funds to invest in this way. We have already expressed sympathy for this argument in our discussion of failures in the market for risk-capital. Whether specially targetted programs should be established to support marketing-related activities per se is another question that we will address below.

One other policy implication of the foregoing discussion might be noted. Given the emergence of economies of scale in marketing packaged software, and the acknowledged scarcity of marketing expertise in Canada, one would expect mounting interest in merging, joint-venturing and other forms of cooperative marketing. Indeed, these developments are accelerating in the applications end of the software industry. We see no useful proactive role for government to play in this regard. Self-interested businesses will presumably find ways to conserve on scarce resources and economize on scale advantages. However, public policy could inadvertantly thwart industry efforts to rationalize capacity through, say, an overly aggressive application of the Combines Act. To date, there is no necessary reason to believe that this is a problem.

One final factor to consider is the presence of multinational companies in the industry. It has been argued that foreign-owned companies do too little R&D in Canada and develop and market very few innovations from their Canadian base. Again, this is an issue that is too complex to treat in this specific study. What is clear, however, is that subsidies to foreign-owned companies represent potential transfers from Canadians to non-Canadian investors. Hence, it is important to ensure that policies designed to promote software development activity in foreign-owned subsidiaries promise equivalent benefits to Canadian residents.

In summary, the recommendation of specific public policies should draw heavily on an assessment of what is going wrong in the marketplace. However, it should be remembered that beyond intrinsic market failures, the extensive involvement of other governments has already distorted "pure" markets for software. Policies which do not address the underlying sources of inefficiency are unlikely to move the sector(s) involved toward greater efficiency. For example, if inappropriabilities are the major source of market failure in a given set of activities, the problem is probably best dealt with by subsidies which can be varied with the circumstances of each project, rather than by across-the-board tax incentives, which are certain to over or under compensate for the inappropriable benefits of any particular project and may err substantially in many cases.[132] On the other hand, where the source of market inefficiency involves biases in the capital markets which restrict the availability of risk capital, and where the projects involved are relatively small and relatively homogenous, the administrative costs of discretionary grants may make tax expenditures a preferable way to stimulate output.

Before evaluating specific policy instruments and offering our own policy recommendations, we would like to suggest a useful framework to categorize alternative public policies. We suggest that public policy designed to promote increased economic activity in a given sector can be characterized along two dimensions:

1. whether the policies operate primarily on the supply side or the demand side of a market;

2. whether the policies are targetted at specific firms and/or specific types of software or whether the policies are generally available to all firms in a given industry.

As an illustration of this theme, supply-side policies are those directed at increasing the availability of critical inputs to software design, development and marketing activities. An increase in the supply of inputs would lower costs, shift the industry supply curve to the right, and encourage an increase in the quantity of software demanded through the stimulus of either better quality and/or lower prices. On the other hand, demand-side oriented policies would be directed at shifting the demand curve for software to the right, say by increased government procurement of locally developed software, or by allowing more favourable depreciation treatment for capital equipment that embodies locally developed software.

In this case, output in the sector would increase through a movement along the industry's supply curve.

If demand in the market is growing rapidly as a result of market forces, additional demand stimulus as a result of public policy is unlikely to have significant marginal impact. On the other hand, if the industry supply curve is shifting to the right at a fairly rapid rate as a result of market forces, public policies that act on the supply side of the industry are likely to be difficult to justify on grounds of cost-effectiveness.

In a similar manner, evaluation of the merits of targetted versus general policies should be sensitive to the nature of the activities involved. For example, where it is difficult for policy-makers to distinguish among claimants for government assistance, it may be more effective for government to adopt broad-based tax incentive programs rather than specific grants. On the other hand, when projects are relatively large and heterogeneous, selective grants may be both a viable and efficient policy.

With this framework, we now turn to an examination of specific software policies.

7.1.3 Canada's Trade Deficit in Software

Canada's balance of trade deficit in computer hardware and software products of all types (mainframe, micro, mini) continues to grow. In 1982 Canadian trade deficits just for office equipment and computers were in excess of $2 billion. Although equivalent exports were almost $1 billion, total imports rose to $3 billion, thus creating a trade imbalance in this section of over $2 billion (mainly involving the import of personal computers and word processors). In 1980 the equivalent imbalance was $1.2 billion.

In 1980 alone, the most recent year for which this data could be gleaned, software exports were $35 million, but imports were slightly greater than $900 million, yielding a net software trade imbalance just for that year of almost $1 billion.[133]

If this large trade deficit represents a concern to policymakers, one possible response is to try to make Canada's future balance of payments in these technology areas depend more directly on software than on hardware production. And if this is the case, national policies and marketing strategies are necessitated which vigorously support the present main thrust of software production in Canada -- the production of business micro software -- and which also offer additional incentives to the rapidly growing sectors of this industry, such as AI software.

Many policy measures will directly or indirectly affect software industrial development, including tax/fiscal treatment and support, R&D policy, export policy, marketing policy, educational/training policy and copyright policy.

The following sections first summarize and critique existing Canadian policy instruments in each of the above areas and then make specific recommendations concerning how they may be enhanced.

R&D in knowledge-intensive industries, for example, is different from R&D in manufacturing in several ways, and as one reads through the intricacies of the existing tax rules and byzantine interpretive tax bulletins pertaining to software, one cannot help realizing that they were conceived from an industrial manufacturing perspective.

In Canada, governmental fiscal/taxation programs to aid software/microelectronics and other high technology industries have been less forceful than those in many countries and, in the case of software, have often been limited to applying R&D benefits, which themselves, when compared to R&D benefits of several countries, could be more competitive. The Ontario Microelectronics Task Force,[134] for example, recommended in 1981 that programming and design of software should be treated as R&D, and that the definition of R&D be expanded to include marketing and pre-production costs. Although one might agree with the thrust of such an expansion, it would unduly distend the definition of R&D to include activities which involve neither research nor product development.

This report also recommended that sales tax exemptions be applied to software producers and that a number of tax/fiscal incentives -- such as the prorating of equipment used in software writing -- accelerated depreciation allowances, and manufacturing investment tax credits be applied to Canada's software industry as they are presently applied to hardware production.

Although we agree with these latter incentives, given the fiscal/taxation support to software industries by Canada's competitors, such measures in and of themselves may not be sufficient to significantly affect the development of Canada's software industry. In a number of other countries, in fact, tax/fiscal measures supporting software and other high technology industries include tax holidays, setting up special tax reserves to support export marketing costs and export financing in general, tying specific tax incentives to the quality and quantity of export performance, low interest loans to special industries, and up to 200% R&D tax exemptions on computers, software and other knowledge industries.

Another reason why substantial incentive measures may be called for is because many of Canada's software companies during the past few years have been making the transition from custom to packaged software, involving massive capital outlays during a recession. Also, as noted, tax laws have been written from an industrial/manufacturing perspective, and software production differs significantly from traditional manufacturing. In an Economic Council analysis of expenditure ratios on 237 new products and manufacturing processes, it was found that average marketing costs as a percentage of total costs to develop and bring the product/process to the market involved only 7.5%, while the marketing start-up costs of software products may involve a third to a half of this total cost.

Secondly, the manufacturing start-up phase of the development of manufactured products comprised, on average, 33.7% of these total costs, while the "manufacturing start-up costs" of software products are often insignificant compared to other costs involved in software design, marketing and maintenance.[135]

It is also evident that just because a policy has been successful in other countries, such as Brazil, is no reason to expect success in Canada, given the significant economic, political and cultural differences that exist between countries. The Brazil software industry, for example, is protected by the physical barriers of the

Portuguese language and a considerable distance from the US. Given Canada's technological dependence on the United States as both buyer and seller of computers and software, and given our physical proximity, adopting a Brazil-type software import ban would first of all not be enforceable under current trade agreements, and secondly even if it were partially enforceable, it would have obvious negative economic consequences, given the likelihood of retaliation by the US.

A number of developing countries, in fact, have invoked protective regulations pertaining to the importation of technology and software, and some benefits have included minimal savings in the balance of payments, and limited growth of indigenous industries. The major costs involved are more severe -- a scarcity of technological goods, technological inefficiencies, while indigenous industries are being built up and significant price increases. Also, developing countries (and Canada) usually have very small domestic markets for high technology goods and services, and any protected market will grow very slowly compared to international markets.

A nascent industry such as software cannot be offered a protectionist shield from imports from our major trading partner, the United States. What is instead required is a strategy that supports the competitive development of the industry, including an ability to export worldwide.

7.2 AI Policy

Although the following policy sections concentrate mainly on instruments which are applicable to Canada's software industry as a whole, obviously there may be policy instruments which are applicable predominately to a single field of software, say industrial automation software. (Accordingly, tax incentives to adopt automation equipment may affect the demand for that software.)

There might also be policy instruments directed to types or classes of software which may be central in many applications, such as artificial intelligence software. Since this topic is the subject of a recent five volume Cognos report,[136] it is appropriate to first review their major findings and recommendations.

The Cognos study first suggests both areas and levels of AI R&D activity in Canada, currently funded at around $5 million annually, and describes some institutional responses for the development of a "critical mass".

The alternatives considered are to opt to have a Japanese style comprehensive national plan or more "piece-meal" altering and expansion of existing institutions. Given the limited Canadian R&D resources, the study chooses the second alternative.

7.2.1 Demand Stimulation

Although it is thought that government funding must initially "carry the burden" of the commercialization of AI, venture capital tends to follow national interests in technology, and given the nature of necessary expenditures, ultimately AI develop ments cannot be supported only by the federal government. To help increase the level of private AI research and development, it is suggested that the Canadian government "advertise priorities" of AI research to attract more research allocation to the field, and that the government initiate "specific mission oriented demands".

It is also recommended that AI R&D activities of the federal and provincial governments be merged via cooperative sub-agreements (easier said than done). Specific areas of mutual research suggested here includes computer-aided instruction, both for education and job training, medical expert systems and machine translation.

Within the area of machine translation, it is suggested that the Secretary of State rapidly develop common R&D objectives with the Department of National Defense and encourage more extensive and rapid AI R&D procurement based on common specifications.

7.2.2 National AI Spending Targets

The study proposes a broad ten-year Canadian AI R&D program with expenditure levels (including all national R&D activities such as private sector contributions, grants, contracts, etc.) as follows (Exhibit 7-3).

There is a doubling of AI expenditures per annum for the first three years and fourth and fifth year increases of 75% and 50%, respectively. Anticipating that the second five years' funding will be predominantly market driven, most of the proposed expenditures involve 20% annual increases, with a total second five-year R&D effort of $535 million.

As the study remarks, these expenditures are not large compared to the equivalent expenditures in other industrialized countries (over $1 billion and $2.5 billion in Japan and the US, respectively) over the next ten years.

EXHIBIT 7-3

Phase One: Target R&D Expenditures on AI

Year	R&D Expenditures Related to AI	Absolute Growth Over Previous Year	Percentage Increase Over Previous Year
		($ million)	
1984	$ 10	$ 5	100%
1985	20	10	100%
1986	40	20	100%
1987	70	30	75%
1988	105	35	50%
5 Year Total	$245		

Phase Two: Target R&D Expenditures on AI

Year	R&D Expenditures Related to AI	Absolute Growth Over Previous Year	Percentage Increase Over Previous Year
		($ million)	
1989	$126	$21	20%
1990	151	25	20%
1991	181	30	20%
1992	217	36	20%
1993	260	43	20%
5 Year Total	$935		

Source: Cognos

7.2.3 Short and Medium Term R&D Goals

Within the short term (two to three years), the Cognos study proposes a research goal of assembling Canadian resources in machine translation, telecommunications and office automation to manufacture a translator's work station; secondly, to explore contractual arrangements with foreign firms to develop in Canada and market throughout the world a French version of NLP products such as AI Corporation's Intellect or IBM's Epistle. The final short term goal involves the development of intelligent computer-aided language instruction systems.

As mid-term goals (three to five years), the study recommends:

1. The development of "sub-language" oriented small-scale machine translation systems, building on TAUM/METEO, which should be extended to handle weather forecasts in either French or English, and some governmental translation.

2. Secondly, to develop a system for "automatic bilingual synthesis of natural language reports and non-linguistic data".[137] Proposed restricted domains include stock market and agricultural market reports.

3. Finally, to develop more extensive French versions of natural language query and text critiquing systems.

7.2.4 Long Term R&D

There remains a number of unsolved basic research problems in the area of natural language processing -- problems in understanding the representation of knowledge, in semantics and in ways people make inferences. In addition to these basic problems, there are further basic research problems in speech and machine translation, central of which is a series of issues dealing with the nature of rules utilized for translation from one language to another. For example, are there general rules to translate a sentence from one language to another or translation rules which are specific to the two languages involved?

Specifically, as a long term research effort the study proposes to:

"1. Assess the possible contribution to MT theory of recent results in sematics, pragmatics and AI as, for example:

- sematic theories such as Montague grammers and situational sematics
- knowledge representation techniques used in state-of-the-art AI systems (such as frames, non-monotonic logic, etc.)
- recent developments of computer science, such as logical programming.

2. Examine and detail the knowledge used by expert translators and try to devise more powerful schemes for translation rules.

3. Integrate the results of 1 and 2 into a coherent framework practical for third generation machine translation systems.

4. Produce a prototype system for restricted domain (such as, for example, the 'synopsis' part of weather forecasts which have been judged too difficult for the technology used in the TAU/METEO system).

5. Apply a similar technology to other NLP applications, such as sophisticated data base query systems." (138)

7.2.5 Summary

In summary, the study recommends that as components of a national AI plan, the Canadian government enact these measures:

"- within the federal science and technology envelope to establish AI as strategic technology and objective;

- in its role as customer, to identify and proceed with contracted requirements for systems, products and services in AI, including in the contracts provisions for the industrial strategy objectives that have been raised;

- to establish and support particular opportunities (in, for in stance, machine translation) where major users and technology suppliers can combine to rapidly develop responses to operational requirements;

- to encourage and support provincial entities in, for instance, education and medical research, in respect of advances in AI technology;

- to strengthen the support from NSERC for strategic grants to the AI community with the twin goals of supporting promising research and encouraging an increase in the supply of highly qualified scientific manpower;
- to strengthen existing initiatives to develop university capabilities;
- to actively support the increase in interactivity between foreign sources of technology and the Canadian AI community;
- to support nascent developmental and marketing joint ventures, on a selected basis, as proposed to the private sector;
- to ensure major research centres exist for special equipment and critical mass of manpower as required." (139)

The study also makes the following specific suggestions:

- to develop a computer network similar to the ARPA network in the United States;
- to use procurement to launch a major project in "second generation" machine translation; this would be done through the Secretary of State and Department of National Defense. Secondly, for the Department of Communications, the Department of National Defense plus other departments to utilize procurement in the area of natural query languages and natural language processing as related to office automation and data bases in various professions, and finally, to use procurement for the development of image recognition and expert systems in sectors such as resources, medicine and education.

The Cognos plan mainly concentrates on three matters:

1. A national spending schedule to support R&D in artificial intelligence.
2. Greater use of procurement.
3. Government's adoption of AI as a "strategic technology".

Although R&D in AI should undoubtedly be rapidly increased, most of the commercially available AI products discussed in this work were based on the availability of publically-funded AI research results in the US, some up to a decade old, and it is precisely in the area of commercializing AI products that several firms have experienced difficulty. Beyond R&D support, software development may require support at the stages of testing, final product design, market research, actual new product marketing and so forth.

We agree with the thrust of Cognos' objectives to prioritize research goals and to make government procurement complementary to other programs designed to promote AI developments in Canada. However, we are less enamoured with the setting of formal R&D spending schedules and the use of government procurement as a direct technique to stimulate AI design and development activity in Canada. The history of direct government management of innovation activities in Canada is not one marked by success. Therefore, we are more inclined to argue for policies that allow for maximum direction of resource allocation decisions by private-sector participants, including the latter's use of resources in universities and the prioritization of objectives by individuals with equity interests in the outcomes of their decisions.

In this regard, we concur with the broad concensus of the software industry, that the best way to support the commercial stages of software development and sales is through a variety of tax/fiscal and export instruments, some of which have been or are being adapted in a number of countries.

We begin with an examination of Canadian tax treatment of computer software.

7.3 Fiscal/Tax and R&D Policy

7.3.1 Research and Development Tax Rules

It has long been argued, with some justification, that Canada's R&D tax deduction schedules discriminate against small and medium sized firms (of all types and not merely software firms) because these cannot afford to have separate R&D operations, requisite to claim present R&D tax deductions. There is also evidence that Revenue Canada's tax auditors are extremely strict in their interpretation of that part of the Income Tax Act which describes rules for R&D deductions.

As presently formulated, Section 37 of the R&D tax rules and Department of National Revenue Interpretation Bulletin No. IT-151-R2 allow a firm to deduct R&D expenses if and only if they are "wholly attributable" to scientific research. For example, allowable expenditures as specified in the Revenue Canada tax bulletin of 1980 don't include "general administrative expenses or factory overhead expenses that would have been incurred even if scientific research had not been carried on."

In effect then, if a firm does not have a physically separate R&D operation which is used exclusively for R&D, it cannot claim any overhead portion of the resources utilized, including costs for equipment, rental, etc., and any other items not included in wages.

Eric Welling, president of the Electrical and Electronic Manufacturers Association of Canada, has noted that tax auditors regularly reject deductions involving R&D expenditures which are not attributable to a separate R&D facility.[140]

> "There is a high measure of uncertainty faced by companies claiming R&D expenses, because they don't know if Revenue Canada will allow the deductions they have claimed, which means they don't know how much money they have to work with." (141)

What this ruling means is that if a firm does not have separate R&D facilities used exclusively for R&D purposes, that firm can make no claims for overhead parts of any resources used:

> "Another way of saying this is that if the facilities ... are used for any work which is not R&D, then no charge, even proportionally, may be made for them when they are used for R&D purposes." (142)

The problem is this: Under the current Income Tax Act, R&D is explicitly viewed as a set of activities which must be physically separate from the "manufacturing processes". This notion is not reflective of Canadian industrial R&D activities in general and is disasterous with respect to the software industry. In software production, research and product development involved in software design seldom occur at "separate facilities" nor is this generally the case across Canadian industries. In fact, in 1977 in a "Directory of Scientific and Technological Capabilities in Canadian Industry,"[143] it was realized that research, development and related activities in any industry may be carried out by either separate scientific and technological groups or by groups which comprise part of an organization's production facilities. Most Canadian companies simply cannot afford entirely separate R&D facilities which are devoted exclusively to scientific research and development, and unfortunately this restrictive interpretation according to which factory overhead is inadmissable as an R&D expense has also been extended to the overhead costs of design and development departments of engineering and software firms.

Since audits are frequently three to four years behind, this interpretation subjects firms (of all types and not merely software firms) to a not yet known level of liability, which dampens any incentives to do any developmental work.

The Electrical and Electronic Manufacturers Association has recommended that "wholly attributable" be altered to "attributable" in the tax law, but the Department of Finance has claimed that multiple allocation of costs for several activities could not be reasonably administered.

In addition, the EEMAC has recommended that:

1. the government change the rules for current capital expenditures
2. the definition of "scientific research" be widened
3. the total cost of R&D be included in legitimate deductions.[144]

7.3.2 Licenced Software

In mid-1983 the Department of Finance restored 100% tax write-offs for firms which utilized applications software licenced from other firms, and other amendments which the Department of Finance is now proposing to the income tax laws are expected to benefit both sellers and customers of computer software.

Before this announcement, the Department of Revenue disallowed any depreciation expenses that businesses often claimed on software they had licenced. The Department of Finance consistently argued that since the copyright had remained with the vendor, the business did not actually own this software but merely owned the right to use it. Given this "vacation condominium" interpretation, firms could still treat software costs as an eligible capital expenditure, but only one half of the cost could be written off at a rate of 10% per year.

(In a previous 1981 ruling it had been decided that firms did not actually own software which they had retained under licence from vendors. As a consequence, the amount that could be written off on computer software was significantly reduced.)

The Department of Revenue has claimed that this reduction arose simply because of the ways software is normally sold, i.e. vendors normally sell only the rights to use proprietary software while retaining the copyright. Given Canada's ancient copyright laws, the vendors want to maintain such rights because there is no other protection. But the Department of Revenue gave a ruling according to which any user must actually own the software (including copyright) to qualify for a complete capital cost allowance.

In a Department of Finance sponsored report,[145] it was proposed that income tax amendments specify that Section 1104(2) of the Income Tax Regulations define computer software costs to include the cost of a licence to utilize software.

Such changes, retroactive to May 1976, now allow 100% capital cost write-offs for a two year period for applications software which was acquired under any licensing arrangement, and a 20%-30% write-off for licenced system software.

In the United States and other countries, in contrast, software vendors have the option to defer any software purchases for five years, in some instances to write-off entire amounts as a current expense.

7.3.3 Software Withholding Tax

Although these new proposals were a step in the right direction, a related issue involves a proposed new withholding tax on payments made to US firms for computer software. In other words, as of July 1983, Canadians who rent or licence via royalty payments the rights to use US-originated computer software are required to withhold a 15% federal income tax. This tax is required on all payments made under a term agreement, but does not affect perpetual multi-national transfers or agreements of unlimited terms, the most common type of licensing agreement.

7.3.4 R&D Partnerships

In an April 1980 White Paper, the Department of Finance attempted to make more attractive tax incentives for limited partnerships and other financial arrangements to increase R&D stimulation.

There was little response, however, from the private sector in 1983 for these high technology R&D limited partnerships.

One of the main barriers to many applications again involved the question of classifying projects as true "scientific research" within the definition of Canada's Income Tax Act.

Although the White Paper examined questions concerning administration of R&D tax shelters, it did not propose any changes to criteria for scientific research nor did it at that time consider whether software development constitutes scientific research.

These financial amendments were directed toward small R&D companies which were so new that they did not yet have revenues against which they could usefully apply R&D tax deductions as then structured.

Lack of coordination in policy, however, has lead the government to simultaneously encourage high technology investment and to enforce restrictive rules concerning the definition of scientific research. William Hutchinson, Chairman of the Canadian Advanced Technology Association, has noted that

> "Ottawa's fiscal activities to promote research and development have been devoid of strategy or direction, and what is minimally required is a federal strategy which identifies and defines specific technologies to be developed and how companies are to pursue R&D." (146)

With a quick slamming of the R&D tax door in connection with these partnerships in 1982, a number of firms were forced to declare bankruptcy because of the considerable finances and time they had spent in getting plans approved for R&D financial support. Several of these were software firms, and one of the few software firms to actually receive the new R&D incentives was Sidney Development Corp. of Vancouver. (Another software firm receiving an R&D tax ruling in 1982, Corvette Data Systems Ltd. of Calgary, went into receivership in February 1983 after it failed to raise the minimum $5 million which had been stipulated in this ruling.)

7.3.5 R&D Flowthroughs - The Scientific Research Tax Credit

More recently, R&D tax flow throughs, first proposed in the April 1983 budget, were passed into law in January 1984 and allow a company to relinquish its scientific research tax credits -- comprising up to 50% of all qualifying R&D

expenditures -- and pass them on to purchasers of new shares or debt. The idea was to make R&D tax shelters available to new (unprofitable) firms which could not use Canada's existing R&D tax credits.

The corporate investor normally gets federal tax credits comprising 50% of the purchase price of any securities, and individuals receive a 34% federal tax credit, which may cumulatively result in 1983 provincial/federal tax credits in the range of 47%-56%. Any issuing company technically becomes liable to a "Part 7 Tax", which is 50% of the proceeds of the finance, but firms can avoid this tax by performing R&D spending which exactly equals these proceeds. Also, if the Part 7 Tax is actually paid in that year, it can be refunded simply by doing a sufficient amount of R&D in the subsequent year.[147]

By March 1984, the Department of Revenue had received filings for more than 400 corporate issuers, which provided a then estimated $685 million in 1983 tax credits.

One of the results of the legislation is that it is being taken up not merely by new start-up companies (which may not claim tax credits on percentages of existing R&D expenditures since they may have none) but also by businesses such as Canadair Ltd., which for example lost so much money in 1983 that it couldn't benefit from existing tax credits which are available on R&D spending.

The Scientific Research Tax Credit was intended to increase Canadian R&D spending, but given the way this incentive is structured, it seems likely that an insubstantial amount of monies raised will actually be spent on R&D.

The main problem is this: After an investor gives, say, $1 to a firm claiming to perform R&D, he immediately gets back 50¢ and a form allowing him to save 50¢ on income tax. But in addition the investor also receives from the company a "bonus premium" of 4% or 5% above his investment. As a consequence, the firm, after it has paid off the investor for the $1 investment, receives a mere 45¢. But to qualify for the tax credit which has just been sold, the firm must do $1 worth of research and development. In other words, if the firm cannot raise the other 55¢ from another source, it simply won't be able to perform the promised R&D and may be liable for tax payments on the money made selling the R&D tax credit to the investor. Since many of the firms using this scheme are cash poor in the first place, a possible consequence may be increased bankruptcies. Another result may be that the government will not be able to collect taxes from firms failing to perform the requisite R&D, and the government may lose close to $1 billion in deferred taxes.

The main effect of this R&D incentive so far, in fact, seems to have been to earn tax-subsidized profit for wealthy investors and foreign multinationals. Although the Scientific Research Tax Credit was instigated to aid firms which pay so little taxes now that they cannot claim significant benefits, this incentive is operated, we have seen, via a middleman mechanism according to which any company can sell its R&D tax credits to other firms or individuals. (An alternate form of R&D incentive might involve a directly refundable tax credit.) However, under the current structuring, Esso Resources Canada Ltd., a subsidiary of Imperial Oil Ltd., for example, paid $70 million to International Electronics Corporation of Victoria, a "one-person firm", which then returned half of this amount ($35 million) to Esso plus the tax credit. But International Electronics Corporation also paid Esso a tax-subsidized profit of $3.5 million as the bonus premium. This is a direct gift to Esso, a multinational subsidiary, and this cost would not have been incurred if the government had utilized directly refundable tax credits. Esso, in fact, has put more than $500 million in these schemes, and assuming it receives a 5% premium, it will have gained a tax-subsidized windfall profit of approximately $25 million.[148]

There are other problems. The incentive is designed in such a way that there is virtually no control over what happens with the money raised (i.e. if any real research is done). The legislation also permits persons to receive R&D funds with no assessment of their technical qualifications. With the quick flip, the investor does not really put up any money and thus has no incentive to check on the legitimacy of the research being "invested in". Indeed, Linda McQuaig reports that when first asked, Esso said they had never heard of International Electronics Corp., only later remembering that they had been involved in a quick flip with this company. Esso spokesperson Kent O'Conner is reported as saying that Esso "has not really studied and is in no way endorsing the research projects in which it has invested."[149]

In spite of the above, the SRTC has worked well in several instances and has done exactly what it was intended to do. For example, several Alberta researchers who started a company named Myrias to make super computers sought funding through the SRTC. However, Ventures West Technology, which put up half of the $1.55 million, sought the help of the head of MIT's Computer Science Department to evaluate the project's technological feasibility, in contrast to the Esso example.[150]

Having reviewed tax initiatives for R&D, let us now examine how the Income Tax Act explicitly applies to software.

7.3.6 The Income Tax Act as Applied to Software

Regulation 1104 (2) of the Income Tax Act has defined systems software as:

> "A combination of computer programs and associated procedures, related technical documentation and data that:
>
> a. performs compilation, assembly, mapping, management or processing of other programs
>
> b. facilitates the functioning of a computer system by other programs
>
> c. provides service utility functions such as media conversion, sorting, merging, system accounting, performance measurement, systems diagnostics or programming aids
>
> d. provides general support functions such as data management, report generation or security control
>
> e. provides general capability to meet widespread categories of problem solving or processing requirements where the specific attributes of the work to be performed are entered as mainly in the forms of parameters, constants or descriptors rather than in program logic."

Applications software is not defined but presumably involves all software besides systems software, as defined above.

In Sections 37 and 127 of the Income Taxation Act, new products and production processes are given certain R&D and scientific research investment tax credits; namely a 100% write-off in the year of choice; a 50% research write-off on any incremental R&D over previous years, and a scientific research investment tax credit. In 1984, the 50% research allowance was replaced by a 10 percentage point increase in the rate of investment tax credit given for R&D expenditures. The revised investment tax credit rates for R&D are:

	Revised Investment Tax Credits
Corporations and individuals (general)	20%
Corporations which receive the small business deduction	35%
Atlantic provinces and the Gaspé, excluding corporations which receive small business deductions	30%

It might be reasonably argued that the development of some computer software involves the creation of a "new product", and that associated costs should qualify for these incentives. However, although some software now qualifies under these incentives, much does not.

Equally important as the issue of whether or not software development costs should be written off as current expenses as occurred or capitalized as the cost of a depreciable asset, is the issue of what types or components of software should be considered R&D?

The Canadian tax definition of research and development is described in Regulation 2900, as supplemented in interpretation IT-439. It defines "development" as:

> "Use of the results of basic or applied research for the purposes of creating new or improved, existing materials, devices, products or processes."

The scope of activities of Regulation 2900 includes software writing, research in psychology and mathematics, operations research and engineering. It specifically excludes normal materials testing and quality control, social sciences and humanities research, market and sales research, actual commercial production of new and improved products and manufacturing processes, and normal data collection.

It would seem then that under these definitions, computer software should qualify as legitimate "development" when it involves development for the purposes of "creating new or improved existing ... products or processes." However, as one study has noted:

> "In this context, new or improved would seem to mean technologically new and improved ... however, (this concept) has proved to be an elusive one historically, both in the US and in Canada." (151)

Noting that court decisions in defining this concept are not useful, this work cites the principal guideline to administrative practice, IT-439, of which paragraphs 16 and 10 read:[152]

> "The designing, building and testing or both prototype hardware and prototype software systems is considered to be R&D. This includes the development of real time and process control systems, and products as well as those programs and systems used to support or analyze activities which constitute R&D in their own right ..."
>
> "A prototype is an original model on which something new is patterned and of which all things of the same type are representations or copies. It is a basic model possessing the essential characteristics of the intended product. The design, construction and testing of prototypes normally fall within the scope of R&D."

If one considers the master copy of the final version of a new software program as a prototype, it would then seem that software should qualify as R&D under these descriptions.

Although Revenue Canada initially considered much software to be legitimately classified as R&D (witness Sidney Development R&D tax shelter offerings), by mid-1983 the Department apparently had second thoughts and issued more restrictive internal guidelines for R&D qualification of software. Probably the most important of these internal guidelines stated that the new software product must not comprise a duplication of any product already on the market or available to the industry. Invoking this restriction, the Department concluded that development costs of applications software did not qualify as scientific research or development.

This clause involving duplication of software on the market is very difficult to determine and involves a misunderstanding of one of the central software market trends, blending and combining hitherto disparate applications in new ways in a single applications package. Besides, applications software might duplicate the functions of a commercially available product and still have better solutions for performing these functions and also have required extensive research. For example, one Vancouver software firm has produced a spread sheet which duplicates products on the market but also incorporates additional features, such as the instant availability of formulae according to which individual entries were computed. This feature necessitated considerable product development.

Another restriction was that software writing had to involve "significant technical risk". However, technical risk has nothing to do with whether or not a new software package comprises a technological advance (and therefore qualifies as research and development). A major cost of software producers involves marketing rather than technical risk, and a more appropriate test might focus on the function of the program. In other words, does the software facilitate new or improved functions or a new integration of existing functions, which has not been previously performed on a computer? If yes, it should qualify for R&D incentives.

Finally, these guidelines also excluded humanities and social sciences, and thus excluded much software for computer-based education and training, one of Canada's main market opportunities.

By late 1983, Revenue Canada had again loosened its rulings and interpretations somewhat, and in October 1983 one company, Microcomp, obtained a favourable ruling on their R&D limited partnership program. (This firm produces both original and highly derivative programs for reservoir modelling in the oil and gas industries to run on microcomputers.)

7.4 Fiscal/Taxation and R&D Recommendations

7.4.1 Capitalizing vs Expensing Software Development Costs and R&D

Tax rulings pertaining to software development costs are set forth in IT-283R (November 1979). In this bulletin, Revenue Canada has described its practices for costing of films, video tapes and computer software.

If the software is required mainly for internal use within a taxpayer's business, this bulletin suggests that it's anticipated lifetime, purpose and nature should in each case be examined to decide whether or not it can be written off in the year in which the development costs occurred, or whether it should be capitalized and written off as the cost of a depreciable asset. But software is to be capitalized when its "useful lifetime" is greater than one year. (However, if the software was acquired after May 25, 1976, and is classified as an actual capital expenditure, the specific capital cost allowance class under which it is classified depends on whether it is systems or applications software.)

This would seem to suggest that all software development costs should be capitalized as the cost of a depreciable asset unless the software has a "useful lifetime" of less than one year.

Furthermore, paragraph 16 of this bulletin states that:

> "Where costs incurred in the development or production of a software program for lease or for the developers' use are capital in nature, the capital costs of these programs will include material, labour and overhead costs that can reasonably be attributed to the production of the property but nothing in respect of profit which might have been earned had the asset been sold."

In the past, although Revenue Canada has generally left it up to firms to decide whether or not software development costs are written off as a current expense at the time occurred or capitalized as a depreciable asset, today this has become a major issue in Canada and the US. On the one hand, the New York Stock Exchange has recently required that development costs be expensed as a condition for software firms to be listed. (This is so investors will not be misled about the true value of the company.) On the other hand, firms in favor of capitalization argue that unless costs are carried, no one knows what the value of the company is. Recently an interdepartmental task force investigating this and related issues has bogged down.

The tax law has been written in an industrial context which may be inappropriate for dealing with the capitalization of information. Also, information products such as software depreciate in value (for the producer) differently than material commodities depreciate. With the exception of antiques, material commodities, say automation equipment, depreciate steadily. With knowledge products such as software, depreciation is often sudden, as, for example, when products are widely copied and revenues are cut in half.

Historically, the Canadian R&D/GDP ratio is lower than that of other OECD countries, and the Canadian preoccupation with increasing industrial R&D levels provides a motivation to introduce tax support of various kinds as applicable to software; possibilities might include exempting:

1. software design costs
2. design plus marketing costs, including marketing research and promotion
3. all development costs for software production
4. only certain types of software, e.g. applications or system software which involves any improvement in the methodology of producing software might be classified as R&D. Finally, development and other costs of systems or applications software derived from, or written in connection with, R&D projects might be exempted.

The Department of Revenue, we have seen, in mid-1983 issued more restrictive guidelines pertaining to R&D qualification of software, the most stringent of which stated that new software must not involve a duplication of any commercially available product, and decided that applications software did not qualify as scientific research or development. But much applications software requires extensive research, and certainly specification and design portions of applications packages should qualify as "scientific research and development", even under the existing wording of the tax law.

Another alternative might be to no longer claim that most writing of applications packages is real scientific research, even the design and implementation phase, and to expand the definition of research in regulation 2900 and interpretation bulletin IT 439 to include applied research.

Interviews with firms have indicated that Revenue Canada could be considerably more flexible in allowing firms to decide whether a specific software development cost should be written off as a current expense as incurred or capitalized as the cost of a depreciable asset in the relevant CCA class. There is certainly a valid concern that arbitrary accounting conventions will distort the effective functioning of the capital markets. But in and of itself, this concern does not argue for one or another accounting convention, as long as consistency is practiced. The basic thinking behind expensing versus capitalizing a depreciable asset after all lies in the notion that costs of developing a product should correspond to the time profile of the revenues generated from it. In this regard, when extensive marketing research has been done on a new software product, there is considerable uncertainty about its useful lifetime, and revenues are more correlated with marketing costs rather than development costs for software. Since firms may vary in the nature of the software products developed, there is some sense in allowing the firms themselves to decide the matter.

We thus recommend that firms be allowed to decide whether software development costs are capitalized as a depreciable asset or expensed. At times there may be good business reasons to capitalize and at other time to expense. This is a decision that firms should be allowed to make. Allowing firms to expense or capitalize at their discretion will facilitate greater control of cash flow and will allow more realistic business planning.

We have already argued against the Ontario Task Force's suggestion of classifying marketing costs as R&D, since this distends the notion of R&D to unrecognizable limits and weakens the incentives themselves. What is necessary is a relatively simple and clear rule for most situations.

We thus recommend that all software development costs of firms whose main business is selling software to third parties should be given the R&D exemptions.

We further recommend that one form of the present R&D incentives -- the Scientific Research Tax Credit -- be eliminated, for all of the reasons previously discussed, and replaced by a mechanism which will directly transfer R&D funds from the government to firms performing legitimate R&D. (Unfortunately this will put the government back into the role of evaluating R&D projects, but at least someone will be performing this role, unlike the case with the SRTC.) An R&D tax rebate should be set up which would permit R&D firms with no taxable income (i.e. many software firms) against which to claim deductions, to exchange R&D tax credits for cash refunds at the beginning of the tax year, and firms with taxable incomes to make the appropriate R&D deductions, with 100% of legitimate R&D expenses exempted.

The appropriate costs incurred in developing software packages which partially duplicate products already on the market, but which add or blend in new functions to old ones, should also qualify for the R&D exemptions. This is a distinction which the United States is currently making in its draft to regulation 1.174-02 IS, i.e. the distinction between maintenance and enhancements, with the former being "mechanical" changes to an existing software package, and enhancements being the adding of new functions to a package or blending in of existing functions in new ways. Maintenance changes would not qualify for the R&D tax rebate, while the costs of new add-ons or enhancements would.

7.4.2 Separate R&D Facilities

We recommend that section 37 of the R&D tax rules be changed so that firms which do not have separate R&D facilities solely developed to scientific research may claim the R&D exemptions. Few software firms in the world have separate facilities devoted to software product development and research, nor for that matter do many firms in other Canadian industries. In software production, "research" is usually more intricately tied to product development.

7.4.3 Licensing Software from Others

We recommend that in the income tax regulations defining computer software to include the cost of a licence to utilize the software, the 100% capital cost write-off presently in effect for applications software be extended to systems software (which presently receives a 20%-30% write-off). The period of write-off should further be increased to six years for both applications and systems software; since software lifetime has been extended by the use of applications generators and other tools, this period should at least expand the useful lifetime of the package, but firms should also be given the option of accelerated depreciation.

7.4.4 The Software Withholding Tax

We recommend the elimination of the software withholding tax of 15% for Canadians who rent or licence through royalty payments the right to use off-shore originated software. If this is an incentive to purchase Canadian software, it misses most of the traffic, since over 95% of the international traffic between Canada and the US in computer software occurs through multinational channels, and this tax, as formulated, would perpetuate multinational transfers and agreements of unlimited terms.

7.4.5 Expansion of the Scope of R&D Incentives

We further recommend that regulation 2900, defining research and development, be changed to include research in education and training. Educational/training software is a major opportunity area for Canadian firms, and it makes no sense whatsoever for this area to be excluded from R&D incentives.

We recommend that the "wholly attributable" clause be eliminated in the R&D provisions and replaced with mechanisms which make costs incurred in a single facility used for production and research proportionately attributable to the overhead and direct costs involved in these two functions. The Department of Finance has claimed that such multiple allocation of costs and multiple activities would create "serious evidentiary and administrative problems". While it is clear that difficult and (in some cases) unavoidably arbitrary imputations will be involved, multiple allocation is undertaken in other countries and in regulated industries in Canada.

We recommend that section 37 and regulations 2901 and 2902 be altered so that capital assets partially used for R&D qualify for R&D incentives in a proportionate way. Excluding related overhead costs from these incentives, unless they are wholly attributable to scientific research, is inappropriate to the way research is done in most Canadian industries and is totally inimical to the way it is done in software firms.

It is difficult to evaluate the fiscal implications of these recommendations, since it is not simply a matter of how existing firms will respond to the changes, but also a function of growth in the software sector. To the extent that sectoral growth generates taxable income, it is possible that no net tax revenue consequences will obtain from the recommendations. The main point here is that the recommendations are consistent with "neutrality", which is another important objective of the tax system.

7.4.6 Tax Holiday on AI Products

Fourth generation software based on artificial intelligence notions, we have seen, is effecting major changes in fields varying from computer-aided learning and training, to systems for industrial automation and natural language interfaces for business data bases. From all available evidence in this study, it is clear that within six to ten years, AI-derived programming techniques will dominate software manufacturing and software products. The Canadian AI expertise is nascent, but most countries don't have any. Therefore, in addition to the previously described R&D assimilation measures for software firms in general, we further recommend that a partial tax holiday for a six to ten year period be given to firms which produce AI software and systems. The basic rationale for special tax treatment for AI activities is our assessment that the technological and commercial "spinoffs" in this area are likely to be quite pronounced over the next few years. That is, appropriability problems and risk are significant for this segment of the industry. A more favourable tax regime will mitigate the disincentives created by appropriability problems, to some extent.

A focus on modifying the tax regime for this sector is suggested, as well, by the fact that R&D costs are merely 10%, on average, of total costs of the whole process of conceiving a new product and bringing it to the market. More important now to software firms are the commercial costs incurred after product research and development, and this entire area is the focus of this incentive. Specifically, we recommend that a minimum 40% of domestic and off-shore sales of AI products and systems be exempted from relevant corporate taxes for a six-year period.

This tax exemption should be placed at the level of actual product sales and pertain to any corporate revenues directly derived from these sales.

With a partial tax holiday on fourth generation products, we are advocating a strongly export-oriented software strategy. The major markets and software opportunities for Canadian producers lie offshore, mainly in the US and Asia. This move would also be a forceful upgrading incentive for producers of traditional software.

This incentive could be structured in a number of ways depending on the deferred revenue which the government is willing to tolerate, e.g. it could apply also to appropriate revenues of Canadian subsidiaries operating offshore along the lines of US foreign sales corporations. These are foreign subsidiaries of US companies whose earnings from US exports are exempt from US taxes (up to a limit). (In this case, one of its effects might be to increase the rate at which Canadian operations are set up offshore, especially in the US.)

Also, the percentage of revenues exempted might vary, depending on how rapidly sales grow and the net fiscal consequences of the exemption. The government would presumably examine and monitor the fiscal consequences of any chosen exemption level and could modify the level accordingly.

We note the possibility of charges by Canada's trading partners that the measure unfairly subsidizes Canadian exports. However, we do not believe this is a significant threat, since the program would be restricted to AI products and would be a terminal program.

We agree that this exemption is open to abuse (as are existing exemptions) and acknowledge both the difficulty of defining "AI products" and the potential revenue loss to the government if this portion of the industry grows as rapidly as we anticipate. AI techniques have been incorporated into, or used to produce, traditional software for a half decade and indeed these techniques are usually transparent to the user; however, given the trends discussed in this work and the emergence of companies whose entire product line is one or two AI products -- front end languages, expert systems, etc., i.e. products whose operation exemplify and incorporate a number of well known AI principles -- the problem of a working definition of "AI products" should not prove insurmountable.

With respect to the second issue, lost revenue, we estimate that in three areas only -- natural language processing, expert systems and visual/voice recognition -- deferred or lost revenue to the government could approximate $300 million annually, assuming a world market of $50 billion by 1990 and exemption of 50% of sales. This estimate ignores the tax revenues gained from the expansion of domestic sectors tied to the growth of AI software production. The presumption underlying this recommendation is that short-run fiscal "pump-priming" will contribute to the emergence of a viable domestic AI-software sector, which is itself a public policy objective.

One other policy adjustment should be considered by government. Namely, a significant liberalization of depreciation schedules for capital equipment embodying AI software. The development of this software will require close coordination between software producers and potential users in a variety of sectors, including the resources sector. A healthy domestic demand for capital equipment is a likely prerequisite to the collaboration needed between developers of expert systems and downstream users. Given the relatively low capital investment rates in Canada, especially in the resource sector, re-examination of depreciation schedules is desirable in any case. Another point to make in this regard is that decreases in the after-tax cost of physical capital will directly stimulate the performance of research and development, since investment in physical capital and R&D capital tends to be complements in production activities. Indeed, at least one study has found that tax incentives for investment in plant and equipment are more effective in stimulating R&D (per dollar of subsidy) than tax incentives operating directly on R&D expenditures.[153]

7.5 Software Piracy, Copyright Policy and the Law of Trade Secrets

By conservative estimate, for each microcomputer program sold in North America, between 10 and 25 illegal copies are made. Microcomputer software piracy is a real problem, with each floppy disk program costing between several hundred and a thousand dollars. Unlike copies of books and films, software copies are just as good as the originals.

Adapso, a United States data processing association, has stated that it considers microcomputer software piracy the most pressing problem which faces the industry. In addition, there is extensive software copying in such countries as Taiwan, Hong Kong and South Korea.

It is presently impossible to legally enforce software copyright laws, and in spite of various ingenious technological attempts to prevent microcomputer disks from being copied, one may expect the same copying trends to occur in the micro-computer software industry as have occurred with audio and visual cassettes.

Thus far a variety of technical methods to stop software piracy have failed. A wide range of piracy programs are sold by the microcomputer software industry itself, including "bit and nibble" mode copiers which scan the computer program bit by bit and transfer it to another floppy disk.

A new technological innovation, however, invented by the Vault Corporation of California to prevent software piracy, involves a "fingerprinted" floppy disk. In a secret method, Vault places a physical print in the disk which serves as a type of code in the disk material. When such an imprinted disk is placed into a disk drive, to run, the program must first locate and read this imprint. If the imprint is not there, the program will not run.

This protection device, being a combination of software and hardware, is impervious to the voracious bit and nibble copiers. While copies can easily be made of any software programs' applications instructions, unless the fingerprint is on the disk, the copied instructions will not be followed.

(It is, in fact, mandatory that any user be able to regularly make copies of any software program, since microcomputers frequently damage disks, and such copying is a standard procedure to make a backup.)

With the Vault Corporation system, when the user firsts acquires an imprinted disk he may copy his software onto an ordinary non-imprinted disk and simply retain it. If it is subsequently needed, he can place the imprinted disk into the first disk drive and allow the imprint to be read and then simply insert the back-up floppy disk containing the copy of the software program. If the original imprinted disk is not so damaged that it can actually rotate and be read by the disk drive, it's imprint will still be read and will tell the microcomputer to execute what instructions have been placed on the copy disk.

Although this imprint system itself can be cracked, any software pirate would have to have very expensive equipment for so doing, and it's utilization should eliminate at least the type of casual home copying which normally goes on with microcomputer software. However, a number of potential users could still share a fingerprinted disk and use copies of the applications program.

The Vault Corporation is planning to sell imprinted disks to software firms for approximately $1 more than floppy disks normally cost. Since the imprinted disks will run on most brands of microcomputers, Vault hopes that their imprinted disks will become an industry standard.

This piracy of software will likely be increased by the growth of computer literacy in the general population as perscoms diffuse, and today many people copy personal computer software, one reason being that most software is so overpriced that good ethics do not always overcome personal advantage. A word processing package may either be purchased for $95 to $550 or copied by pirate software (itself copied), which initially retailed for $23.98.

Thus far neither Canada nor the United States is prosecuting individuals for home copying of software or for the possession of illegally copied software programs. In the US, however, there are cases of prosecution of commerical pirates, given the recently updated copyright laws in that country. In Canada, however, the copyright law has not yet been updated. Software copying and piracy is continuing, in the meantime, on a huge scale.

One Canadian company has estimated 60% of its potential revenues were stolen, and they lost sales in 1983 which involved $3 million.

7.5.1 Business Pirating

By mid-1984, software pirating has become more prevalent amongst large corporations. For example, Lotus Development Corporation has recently sued one of its customers, Rixon Inc. (a subsidiary of Schlumberger Ltd.), for allegedly having made unauthorized copies of Lotus 1-2-3 and distributing them to their branch offices.[154] Rixon agreed to an out-of-court settlement and returned all of its copied Lotus programs plus an undisclosed amount of money.

In protective moves, the US software industry has been pressing for tough anti-piracy legislation, and current US federal copyright provisions impose criminal penalties of one year in prison and a $10,000 fine for copiers. Amendments have been introduced in 1984 that would increase this penalty to a five year sentence and a fine of $250,000, applying only to copies for resale.

One alternative here is for the software vendor to simply diffuse the effects of piracy by including bulk discounts in sales and licensing agreements with customers. A licence fee, for example, might include provisions for an unlimited or fixed number of copies to be made.

The software piracy problem may be tackled at both the technical and legal levels, but from a technical perspective, we have seen, solutions are as yet unsatisfactory.

Legally in the US, changes to the copyright laws have made it considerably easier to sue software pirates. Unfortunately as soon as companies are given favourable judgment against software pirates, the latter go bankrupt. For example, in 1981 Micropro, a large US software manufacturer, was awarded $250,000 plus legal costs due to piracy, but has not been able to collect a penny of this since the pirate subsequently went bankrupt.

In Canada, the copyright law (written between 1921 and 1924) protects mainly tangibles such as designs, specifications and descriptions. It is applicable to player piano rolls but not video games. This law also protects software programs in readable form, but does not pertain to programs operating in a computer, existing merely as electronic impulses in circuitry ("embedded" software).

Presently, Apple Computer Inc. of California is suing both Software One Ltd. and Computermat Inc. of Toronto over copyright infringement of embedded software. In a related development, Space File Ltd. of Toronto has sued Smart Computing Systems, also of Toronto, charging that the latter sold copyrighted programs which were pirated by a former employee of Space File. In late 1983 the judge in the case granted an injunction against further sales, claiming that the alleged violation fell within the mandate of the 1924 law, since the software had been recorded on diskettes and therefore had a material form.

This issue of software copyright has become so important it has become a centre of battle between Japan's Ministry of International Trade and Industry (MITI) and the US government. Attempting to accelerate the development of Japan's software, MITI has recently proposed to protect computer software under Japanese copyright laws for up to fifteen years. (Like Canada's, the Japanese copyright laws do not presently protect computer software.) Secondly, MITI proposes to grant itself the authority to force any firm to licence its software to another firm if MITI deems such licensing is in the national interest, or if the other firm has substantially altered any software package and intends to resell it as a new product.

The US has claimed that these moves are an attempt to help Japan catch up with the US software industry by legalizing software pirating. From the Japanese perspective, they are merely replicating a variant of a policy stance they took in early licensing agreements for chip technology, namely controlled competition.

In the US, software is presently protected by copyright for 75 years after its publication, and the owner of the copyright possesses the sole right to grant licences to others. US analysts envision the possibility that if the MITI proposals involving compulsory licensing go through, many developing countries will pass similar legislation which will allow them to expropriate offshore software.

It is clear that these recent moves by MITI have been partially motivated by IBM's attempt to protect its software in Japan. In recent decisions, in fact, Fijutsu Ltd. and Hitachi Ltd., two of Japan's largest computer manufacturers, have agreed to pay large sums to IBM for having copied large amounts of IBM's software, modifying it and selling it to Japanese customers.

7.5.2 The Taipei Decision

In related developments, in March 1984 a Taipei court sentenced six computer executives to eight month prison terms for having copied embedded software of Apple Computer Inc. This is the first time that computer piracy has received jail terms in Taiwan, and most of Apple's 35 other copyright and patent infringement suits around the world have thus far resulted in suspended sentences. Specifically, the Taiwan executives were convicted of illegally copying two copyright programs of the Apple II -- Autostart ROM and Applesoft Basic.[155]

Until now there have been no questions as to whether the copyright laws of Taiwan actually cover software programs or data embedded in computers.

The defendents in this case are appealing the conviction on the basis that Taiwan's copyright provisions protect only printed words, and in-progress revisions of Taiwan's copyright laws pertaining to software reportedly will protect software as a literary work.

Most previous decisions, for example the Apple 1983 decision in Australia, have ruled that software programs contained in read-only memory are not protected by national copyright laws. One exception to this trend is, of course, the United States. The US Court of Appeals ruled in 1983 that both operating systems and computer software embedded in ROMs could be copyrighted. This decision overturned a decision of the lower courts which had denied Apple's injunction against Franklin Computer Corp.'s Apple-compatible computers.

7.5.3 <u>The Law of Trade Secrets</u>

As an alternative to copyright, attempts to achieve patent protection for software or firmware have generally failed.[156] The extent to which patent protection may be applicable to software is not yet clear, but both fees for patent applications and patent execution are very expensive and involve long delays.

Copyright changes have been equally confusing. It has never been clear whether in the US the constitutionally-based copyright law has actually protected computer software programs, but in 1980 one member of a commission to study the new technologies concluded that viewing computer programs as works that could be copyrighted was incompatible with US copyright law, which he argued was intended constitutionally to protect only "writings" and not sets of mathematical algorithims.

By March 1983 only the US, with its new copyright law, and the Philippines, extended copyright protection to computer software, embedded or otherwise. The 1976 US Copyright Act extended the first explicit statutory protection for software, but the Act removed copyright protection from "ideas, procedures, systems, processes or principles", and merely "expression" and not ideas was protected under this copyright law. In other words, when a writer used copyright and registered his entire program, under the law there was always the risk of loss of protection for the idea.

In the early years of US computing, software writers often looked around for other forms of protection. They discovered an area of jurisprudence which had been designed to protect all types of information -- the law of trade secrets - which essentially offers legal protection to virtually any information used in one's business that is not generally known in that business, and is used secretly for competitive advantages.

Legal protection in the law of trade secrets also involves the protection of other firms in special relationships to the owner of the information from disclosing it, except when authorized. The trade secret law, in other words, offered an early form of protection for computer software. It is even more useful in the US than copyright protection, since the latter protects only a specific form of expression but not the underlying concept or idea, while protection via trade secret pertains to both the idea and the expression.

However, a real question remained. Could software be marketed commercially and still legally be viewed as "possessing" the necessary secrecy? (If it could not, then the law of trade secrets was not suitable.) Thus far the US courts have ruled that the commercial marketing of computer software is consistent with software being legally considered as a trade secret, if and only if the users are "bound to no further use or disclosure".[157]

In subsequent court interpretations, it has been found that trade secret law:

> "... provides no exclusive rights as against third parties, with only limited rights to prevent unauthorized use and disclosure by third parties standing in a special contractual or confidential relationship to the trade secret owner. All others are at total liberty to develop independently." (158)

More significantly, in one case[159] the court unequivocally ruled that the trade secret law is not pre-empted by copyright law, with the US Supreme Court giving its official approval by declining to review this case.

Given then that the laws and claims involved in trade and copyright protection are not mutually inconsistent, software writers in the US are increasingly utilizing both copyright and trade secret law as protective means. Software protection as trade secret better guarantees the protection of underlying concepts and ideas as they diffuse through contractual licensing and joint venture agreements. In addition, copyright of software extends protection to third party users who have not signed agreements which restrict disclosure or use. Also, trade secret protection will extend protection to those countries in which copyright protection is limited and is thus particularly applicable to international licensing agreements. Although many developing countries do not have workable copyright legislation, most of these countries explicitly recognize trade secret clauses in licensing agreements.

Another possibility is to separately extend copyright and trade secret protection to differing types of software. Copyright protection may be more appropriate to software which uses mass distribution, and trade secret protection to other types of software.

7.5.4 Copyright Revisions

Under the proposed Canadian revisions, copyrighted material in Canada will be protected regardless of the medium of presentation -- whether information is on floppy discs, silicon chips or magnetic tapes, the unauthorized reproduction will be illegal. A work which has first been "fixed" in a computer data bank will now be accorded copyright protection regardless of whether or not the information has been also fixed in another medium.

Canada's new Copyright Act will also contain special forms of protection for computer programs which are in machine readable form. Also, computer programs which are in human readable form will continue to be protected, as they are now, with the one exception being that any user will be permitted to make a machine readable program from one which is human readable before a copyright's expiration date for that program.

7.6 Copyright Recommendations

Canada's copyright law is so outmoded with respect to software that persons wanting to protect original software programs have attempted to classify these as "machine readable equivalents" of artistic works. Although recent developments in the US have extended copyright protection to embedded software and read-only memory chips, Canada has not.

In general, it would seem these options are available to the federal government:

1. to exclude computer programs (embedded or otherwise) from copyright protection and to exclusively rely on a law of trade secrets;

2. to extend traditional copyright protection to all forms of software;

3. to grant differing types of protection to different sorts of software, utilizing both copyright and trade secret provisions with special attention to machine-readable works.

Both extremes have their problems. Strict refusal to grant any software copyright protection would seriously damage the industry, since producers might not be able to gain profitable revenues from sales. However, applying traditional copyright laws involves this problem: If software programs are explicitly included under copyright law, conventions to which Canada is a signator would necessitate that programs be protected for the lifetime of the writer plus fifty years, i.e. many magnitudes greater than the product life cycle of the software program; however, more seriously, other conventions to which Canada is a signator require that signing members treat foreign works exactly as domestic ones, and since Canada now imports much of its computer software from the US, enforcement of foreign copyright might cause serious damage to attempts to build a domestic industrial base, since many of the Canadian software companies began in association with products of foreign firms. On the other hand, the absence of copyright protection in Canada might discourage joint-venture and other cooperative agreements between Canadian and foreign-owned firms located in Canada.

Although clearly legalistic measures are not sufficient to protect software, we recommend that both copyright and trade secret provisions be applied to software, emphasizing the former for protecting commercial software which is mass distributed and the latter for other types of software, such as that involved in off-shore contracts.

We agree with the proposed copyright revisions that:

1. software be protected regardless of the medium of presentation; and

2. a program which has been first fixed in a data bank be accorded protection, whether or not the information has been fixed also in another medium.

We also recommend that before passage of copyright legislation, the government scrutinize the net social/industrial benefits of specific periods of protection. Assigning differing periods of protection will have differing effects. For example, if existing copyright protection is extended to computer software, periods of protection might vary from the lifetime of the author plus 50 years to merely four or five years. But, if many programs have useful lifetimes of less than 10 years, and some are periodically maintained and enhanced to the point where the original product is unrecognizable, longer terms of protection may extend no significant social benefits. Moreover, attempts might be made to use copyright infringement as a threat to dissuade competitors from entering the market with substitute products, which constitutes another worry about a too generous period of protection. Fixing the "optimal" term of copyright is extremely complex, as noted by one recent study:

> "Comparing the term of protection with the (present) durabilty of software is, at any rate, an incorrect approach to setting the term of protection. The optimal term depends not on what the current state of the art is but on what it will be or could be in the future and on what incentives society wishes to create." (160)

In other words, a shorter period, say of five to ten years, would motivate firms to utilize trade secrecy more actively instead of copyright protection to maintain control after copyright protection has expired, but a longer term might create a greater diffusion of copyrightable software.

7.7 Skills/Training Policy

In 1984 Canada will produce around 3,000 computer graduates, with a shortfall of 2,000. These figures include computer graduates of all types and not just programmers. Programming skills, in fact, may vary from trivial to more sophisticated capabilities in higher demand. Much traditional business programming, for example, can be done by anyone with a high school math degree, while vision software in industrial automation may require decades of training and experience to produce.

Although there is presently a market glut for traditional business programmers, more sophisticated programming skills are in short supply, and the glut of business programmers is itself a short-term effect of the recession.

One of the main policy problems in Canada is that the federal government has few levers through which to influence educational practices. This is especially unfortunate for computer-based learning and training, since in no other area now is educational and industrial development more closely intertwined. All of the newly-industrialized countries of the Pacific Rim integrate industrial and educational policies. Canada has not. We estimate that over the coming decade, Canada will minimally need 23,000 new computer scientists, programmers and systems analysts, most of whom will be employed in traditional industries, and Canada's present software training output has yet to be evaluated with respect to (even short term) industrial development needs.[161]

In addition to assessing training needs for new programmers, systems analysts, etc., policy makers here must be aware of the export potential of educational/training software, discussed in the export section.

The main problem in implementing any policy recommendations in computer-based training and learning involves federal-provincial relations, and in a forthcoming IRPP book,[162] the authors have discussed how federal-provincial relations in Canada have contributed to lack of development in educational computing.

Federal support for education occurs mainly at the post-secondary level, and comprises over half the costs of post-secondary education in Canada (Exhibit 7-4). The main forms of federal support involve the Established Program Financing (EPF) arrangements, unconditional block transfers of funds that amount to over $3B annually. There is a real lack of provincial accountability for these funds, and as sums have grown, the provinces' support for post-secondary education has not.

7.7.1 Canada Employment and Immigration Commission (CEIC): Skilled Manpower Training

CEIC spent over $1B on industrial training in Canada in 1983. The CEIC directly allocates money to industry to encourage training and through the Canada Manpower Training Program (CMPT) buys "seats" for unemployed individuals in training classes.

Though the federal government supports hundreds of training courses through the CMPT, it does so through provincial intermediaries in a way that reflects provincial jurisdiction in education. There has generally been no direct way for the federal government to fund training courses, and at times this had made curriculum planning difficult.

As computerization in the workplace changes jobs or displaces workers, an enormous amount of retraining will be required in the coming years. Various means to stimulate training in industry have been suggested, and the federal government has recently expanded its own role.

In 1982 the federal Training Act was passed to establish a training program in Canada. One component, a Skills Growth Fund, provides capital for training facilities for "occupations of national importance" where the development of skills is of special significance. The CEIC can now enter into an agreement with either a province or a non-profit organization to give financial aid for training facilities -- buildings and equipment, initial operating costs and development of courses as well. This direct role in curriculum development is a change from the past.

EXHIBIT 7-4

Federal Expenditures and Transfers Related to Post-Secondary Education ($'000)

University Education	1979-80 (actual)	1980-81 (budgeted)
Operating Grants:		
Military colleges	35.1	38.4
Other ministries: operating grants	16.3	17.8
Grants for sponsored research:		
- Department of National Defense to military colleges	0.7	0.8
- National Health and Welfare	9.2	1.3
- Environment	0.8	0.8
- Natural Sciences and Engineering Research Council	104.0	137.1
- Social Sciences and Humanities Research Council	14.6	17.4
- Medical Research Council	58.7	72.4
- Research grants from other ministries	46.6	42.5
Capital grants	0.9	1.0
Scholarship, student aid*	56.1	74.7
Cost of loans to students	47.1	63.0
Other ministerial expenditures	38.3	30.8
Total	428.9	498.0
Transfers to provinces for:		
Post-secondary education (EPF)	2,775.5	3,074.8
Minority language programs	189.7	205.1
Total transfers	2,965.2	3,279.9

* Scholarships, bursaries, awards, student aid from many federal ministries and agencies.

Computer education could be encouraged by federal training policies in two ways: computer-based training could be emphasized in federal programs that fund industrial training and could be implemented in training centres which use federal funds. More generally, the Training Act acknowledges that there are serious training problems in Canada and gives the federal government a broader planning role. No matter what mix of federal and provincial agencies are used to deliver training, long term planning for training should include the use of new technologies such as AI-based computer learning to help meet new demands.

7.7.2 CL Software Procurement

The federal government also spends vast sums on training the armed forces and the public service (Exhibit 5-10). Government can act as a major client for computer-based training technologies and software, exploiting the computer's strong points such as simulations, reduced learning time and intelligent tutorials.

In the US, numerous military agencies have played a prominent role in developing computer-based technology, such as interactive videodiscs, and have supported basic research and advanced developmental projects. Expertise has been built up in a number of computer learning and software firms through military contracts. The Canadian military, a more modest institution than that of the US, spends $600M a year on training. There has been some small-scale experimentation with computer-based training in the Canadian armed forces, but the sole major commercial contract to date was awarded to a US firm. Neither the military nor the much smaller training divisions in the public service have supported computer-based training in Canada to any significant degree.

The provinces are also active in computer learning/retraining, and Exhibit 7-5 summarizes provincial support measures.

EXHIBIT 7-5
Provincial Government Actions Towards Computer Learning (January 1983)

Province	R&D	Indexing, Clearinghouse, etc.	Teacher Training	Procurement Policies	Financial Support
BC		Indexing project (several provinces participating)* Some MECC** material acquired	KNOW (educational TV network): course on micros for educators		100 Apples piloted in schools Some exemplary courseware
Alberta (3 year computer technology project)	ACCESS (educational communications authority): considering computer-based multi-media training system; Electronic communications network	Indexing project* Clearinghouse project: evaluating Math 1-12; exploring means to acquire and distribute content ACCESS: potential clearinghouse role for post-secondary material	In-service project ACCESS: broadcasts aimed at educators; multimedia computer literacy workshop package	Purchase of 1,000 Bell & Howell micros to resell to schools (for standardization)	Discounts for approved and recommended content (40%, 15%) ACCESS: some CL modules made
Saskatchewan		Saskcomp (crown corp for computers): provides a MECC membership to schools; Could market the content of Saskatchewan authors	Guide to micros	Saskcomp: stocks and sells Bell & Howell micros	
Manitoba	Two small pilot projects	Ministry is an institutional member of MECC A consortium of school divisions: in servicing, training, content development and distribution			Ministry of Education partly funds consortium Also some content development
Ontario	Intelligent videodisc project Educational micro Impact of LOGO study OECA: Telidon & educational project; student online database; Telidon downloading	Indexing project* OECA: software of a clearinghouse	OECA: "Computer Academy" TV broadcasts and correspondence course about microcomputers	Developing functional specs for educational micro Copyright arrangements for Ministry-supported content	Subsidized purchases of approved educational micro Exemplary software Major content support (1984)

EXHIBIT 7-5
(continued)

Province	R&D	Indexing, Clearinghouse, etc.	Teacher Training	Procurement Policies	Financial Support
Quebec	Doctoral program in Ed Technology at Concordia: R&D in tele-ed, industrial training	Projects to catalogue & evaluate content and build a bank of "didacticiels" (i.e. CL applications)		Development of educational microcomputer (1984)	Some support for content development
Maritime Provinces			Investigating use of TVO material		

* BC, Alberta, Ontario and NWT are participating in a project on the indexing and cataloguing of CL material.
** Minnesota Educational Computing Consortium.

7.8 Education/Training Recommendations

It is recommended that the government scrutinize the necessary means to ensure that Canada has sufficient personnel for software production and education. The government should evaluate the quality of existing training and education in software production and assess the relationship of the output of these programs to the job market over the next ten years.

Using the US Research Cooperatives as a model, we also recommend that the Department of Revenue investigate and adopt tax incentives which make it more economically attractive for firms to invest portions of profits in university research centres, such as Waterloo's computer science department.

The above assessment should be made taking into consideration the trend toward automation of software writing and the necessity for more sophisticated training.

Although strictly speaking there are no completely automated programs on the market, there is, we have seen, semi-automated software. This trend toward semi-automated programming will slightly decrease the software labour demand around 1990 as semi-automated software design and maintenance tools increase programming productivity at levels significantly higher than the current annual increases of 4%. Someone, however, will still have to write automation programs and even with these gains, software skills will continue to be in demand, especially the more sophisticated skills.

Fourth and fifth generation software, deriving from AI approaches, will be the main driving force in the world's software industries within a decade, and educational/training assessments and matching of training with industrial and social needs must focus now on these new developments.

Existing graduate study programs in computer software should be maintained and enlarged, and special incentives should be considered to private-sector firms that will engage in sophisticated software training of Canadians. Canadian computer science departments are so understaffed that several have been using accounting programs as a screening device for students to get in.

Directly related to issues of software education and training in Canada are those issues pertaining to the development of Canada's CL industry, and the fact that the federal government presently has few policies levers in education.

Given the lack of direct federal influence in the area which creates a policy gap detrimental to industry, it is recommended that a governmental branch devoted to computer learning and training software and hardware be set up. Such a branch would be most appropriately placed at DRIE and must involve persons with trade and export expertise. If such a branch were placed at DRIE, its role must not be limited to concerns of industrial development, although such development should be the main focus; what is suggested is a major effort of guardianship and promotion as has occurred with the office automation project at the Department of Communications.

At the same time, the new branch could appropriately administer a grants program for exemplary projects in "smart" computer-aided learning and give advice on NSERC and SSHRC grants for CL. There is a coordinating role to be played as well, where those who are interested can find information relevant to the CL industries -- market information, names of Canadian firms, provincial educational policies, and so on.

The development of export markets should be emphasized by the recommended new branch, and in particular the educational microcomputer of the CEM Corporation could be used as a vehicle for sales of Canadian training/educational software throughout the newly industrialized countries of the Pacific Rim (whose industries have expressed considerable interest in computer-based training) and elsewhere. Ontario is spending tens of millions of dollars on software and courseware for the Icon, and much of this has resale value in the US, Asia and Europe.

Canada has a small CL market and is highly permeable to imported US software products. A variety of mechanisms can be brought to bear to encourage expansion into export markets -- favourable trade agreements, financing aid, tax support, market intelligence, information on how to do business internationally, representatives abroad to provide information and contacts, etc.

Export potential should also be considered in awarding funding grants to firms, with greater attention being paid to markets outside of the traditional trading patterns of the past.

We recommend that government departments which support substantial amounts of training use CL software and hardware to deal with increasing training demands, and where CL is used should support Canadian industries more vigorously through procurement. In particular, Employment and Immigration, in the new roles created for it by the recent Training Act regarding training in institutions, should use computer-based training. CEIC could also encourage the use of computer-based training within firms themselves, through its programs of support for training in industry.

Obviously, these grant and support recommendations have fiscal implications; however, we believe that much of the requisite funding can be realized through a reallocation of existing public-sector budgets for training and education.

A federal program of support for exemplary projects in CL, such as Ontario's software program for the Icon, should be introduced by the proposed CL branch. The federal government has a history of involvement in educational technology, but there is no focal point at present for this interest, nor is there any funding mechanisms to support educational projects, including computer learning. Projects should take place with provincial cooperation; however, without a federal funding mechanism, such projects seem unlikely to occur. They are most important in the post-secondary setting where there is little activity in CL but where expertise must be built up (in instructional design, authoring, creating advanced CL software, etc.) that can be drawn upon by a growing CL industry.

NSERC and SSHRC (Natural Sciences and Engineering Research Council and Social Sciences and Humanities Research Council) should establish strategic grants programs to fund research on AI-based computer learning in universities, and cooperatives projects between the two councils should also be possible. The emphasis should be placed on both a number of short term needs (in portability software and authoring systems, for example) and longer term research in artificial intelligence-based software.

Provincial ministries should emphasize content development projects in both training and education, and not only attempt to organize the marketplace for CL.

Also, given the fact that it is widely reported by software firms that it requires a couple of years for new graduates to produce work truly useful to the firm, we recommend that steps be taken to establish an accreditation/articling program for systems analysts and programmers. Several countries have such programs, and the government should make appropriate funding available to the several regional software associations now coming into existence in Canada and to CIPS for this purpose.

Finally, we recommend that the government drop current suggestions to adopt a standard (third generation) authoring language.[163] By the time this is implemented, it will have been made obsolete by fourth generation applications in CL, and the adoption of such a standard now will stunt CL development in Canada.

The failure to actively support CL reflects a narrow understanding of major changes occurring now in education and training. How will the widely expected new demands for training be met? Few Canadian training planners or industrial strategists are thinking seriously about the role that new microelectronics software and hardware could play (though this technology is, in fact, a causal force generating new training needs).

Overall, there is little sense of the role that computer learning is likely to play within a few years. It is expected that work will be more computerized; there are studies of employment impacts and programs to stimulate industry in office automation. Is it not likely that learning will be computerized as well? But there is scant attention to CL in the present policy content, and the CL industry finds:

- inattention to CL by planners and by major potential government users of CL (such as the military or Employment and Immigration)

- a lack of aggregation in provincial education markets, which reflects not only the autonomy of education institutions but also the absence of coordinating mechanisms

- a blind spot in funding for either computer learning R&D or developmental projects.

Ontario, meanwhile, stands out in considerable contrast to this overall pattern. Progressive industrial planning has given attention to CL hardware and software, industry stimulation, software exports and to advanced R&D, and planning has occurred to use procurement in schools to develop Canadian educational hardware and software. While some have criticized Ontario's concentration on microcomputers in schools, nonetheless the province is remarkable for its efforts, given the lethargic policy environment for computer-based learning and training in Canada, and Ontario can be expected to reap industrial gains.

7.9 Export and Marketing Policy

In spite of considerable rhetoric about federal policies to aid high technology firms in the developing markets of the US, Asia and other parts of the world, at the working levels there is still a myriad of minor export hindrances and problems, mainly arising from customs and excise. To give just one example, several software firms sell in the US for machines other than that for which the software was written. To see how easily the software may be modified for a new computer, potential software customers will ship the computer to the Canadian software vendor for a few months to research the problem. However, in several cases, customs has indicated that the Canadian company must pay full import price plus duties (and be reimbursed a year later). This puts an unrealistic financial strain on the Canadian firm.

However, in spite of these "minor" problems, with the exception of Brazil's software import ban, major international tariff and non-tariff barriers to software export and trade are presently minimal, and several newly industrialized countries of the Pacific Rim, we saw, have even set up aggressive economic incentives for software firms to open subsidiaries in their countries, develop and export software products.

7.9.1 Export Markets - The US and The Pacific Rim

As the world's software industries mature, these barriers will rapidly grow, and the time to enter new export markets is now. Although the US is, and for several years will remain, our number one trading partner in computers, software and services, of special market interest to many parts of Canada's software industry are the newly industrialized countries of the Pacific Rim and China. For example, throughout the Rim countries many billions of dollars have been budgeted for large scale development projects in resources and communications. To give one example, collectively in New Zealand, Indonesia and Malaysia, $16 billion has been budgeted over the next five years for resource development projects; throughout the Rim countries there is considerable interest in, and plans for, computer-based learning and training within large companies, and China is using computer learning to rapidly absorb western business and management skills. In all of these projects, Canadian sellers will benefit predominately not through hardware sales but through providing the engineering, educational and consulting services, the resource, telecommunications and training software involved in these projects. Much of this work is highly customized.

The potential Chinese market for software and technology dwarfs the combined markets of the rest of the developing world. In the Twelfth Party Congress, the Chinese Communist Party (CCP) called for a GNP of 2.8 trillion yuan ($1.4 trillion US). This can be achieved, the CCP claimed, with the present industrial growth, averaging 8.1% per year from 1953 to 1981, and the Chinese have consistently promised to surpass the West in scientific/technological achievement within the next decade.

But the present reality of China is quite different. There are now 1,008,175,288 people living in China and an additional number equivalent to the population of Canada living in Macao, Hong Kong and Taiwan combined. Of this total population, there are now less than 1 million engineering/technological workers, and most of these are in metal working. A quarter of China's population is illiterate, and China has yet to solve its major agricultural problems; although feeding itself for the first time after the CCP came into control, China has consistently shown a lack of concern for ecological problems, and sustained pesticide use has created an acute shortages of arable land.[164]

Even so, China is making giant strides in both industrialization and "informatization", and since Dang Xiaoping took power, the government has been diverting investment away from the Stalinist lobby involving heavy industry, large steel projects, etc., towards such matters as improved communications and education. Obviously the growth in market demand for various types of software in China depends on this rate of "informatization", but this is now being accelerated.

Beijing is lobbying hard to be able to buy information technologies from the US. Under its 1981/1985 Economic Plan, China has budgeted almost $50 billion in communications, transportation and energy, and the Chinese have been lobbying for quick approval of export licences for items ranging from computer disc drives to satellite-earth stations and scientific software. They are also buying mainframe Honeywell computers for their universities and recently concluded a joint venture agreement with a Singapore consortium to produce microcomputers.

Again, it is apparent that like other Asian countries, China has been evaluating technologies, including software technologies, for incorporation into their industrialization efforts. Although they are sourcing (and copying) traditional business software from well-known American vendors, they are looking to other countries, including Canada, for software and services involved in telecommunications, transportation, agriculture, education and energy development projects.

Educational and training software has promising export potential in schools, industry and consumer markets in both the US and Asia, implemented on micros or minis. Throughout the world there is a demand for training in new computer skills -- how to use microcomputers, how to use new integrated office equipment, how to write software, etc. It makes sense for this training to be computer-based. In industrial training, there are trends towards "generic" modules, suggesting a resale potential for courses developed by a firm for its own training, and which also suggest that a packaged CL software market will emerge as 32 bit perscoms diffuse. Canada can draw on expertise to create training applications in specific areas, such as the use of equipment in the resource industries. There are immense needs internationally, in less industrialized countries, for training in the use of new equipment, and CL could serve as a learning tool. There are also opportunities for tutorial tools which aid in the use of traditional business software and applications generators.

In our discussion of the Canadian software industry, we saw that customized software production is still proportionately higher than in the US, and this fact is thought to be a weakness, given the economies of scale in package production for vertical markets, and the general prediction that soon the world's software industries will be dominated by a few large packaged suppliers, with small firms scrambling for market niches. However, much of this customized production lies in fields such as resource sensing and resource development and management software in the forestry, petroleum and petrochemical industries, telecommunications and training software, and these are precisely the opportunities opening up in the Pacific Rim.

7.9.2 Canadian Software Exports

There is something peculiar about the export of software. If one makes a master disk, it can be mailed offshore and a million copies can be cheaply made for local distribution. This works fine with packaged software, but export of custom software always requires continuous personal interaction and being at a specific customer's site during development.

Packaged software of wide utility is analogous in some respects to books. Recently the Chinese purchased $5 million of US scientific and engineering books on the US East Coast. They bought one copy only of each work. When asked why, they replied "We have Xerox machines." They also have personal computers.

Software may be exported:

1. embedded in equipment
2. through licensing agreements
3. through direct sales
4. through management, consulting or maintenance agreements, service contracts and so forth.

Thus, it is difficult to obtain reliable data on software exports from Canada.

However, Exhibit 7-6 assumes that the Canadian industry's exports as a whole are proportional to those of the software industry in BC. (This is undoubtedly an underestimation.) According to this assumption, Canadian software exports offshore involved less than 8% of the industry's total revenues in 1982. It is only large firms which presently have significant software exports offshore to non-US areas, and these firms also dominate software exports to the US.

EXHIBIT 7-6

Software Exports from Canada as a Percentage of Total 1982 Software Revenues

	All Firms (74) % of Revenue	Large Firms* (12) % of Revenue
Canada	92.6%	79.8%
US	6.5%	12.2%
Non-US	0.9%	8.0%
All Areas	100.0%	100.0%

* Greater than 15 employees.

It should also be remembered that in the BC study it was found that the total revenues from offshore software sales of six large firms were several magnitudes the offshore revenues of all other firms, and that firms with annual revenues of less than $1 million are not operating in international markets.

To lend further significance to the meaning of these figures, it should be noted that the US market for software is roughly equivalent to 17 Canadian domestic markets, and the California market alone is larger than Canada's.

The export figures of Exhibit 7-6 are reinforced by other sources. DRIE has estimated that in 1980, Canadian software exports were $35 million, 7.6% of that year's total software revenues of $457 million. But in that same year, software imports exceeded $900 million (approximately $570 million for imports of bundled software and $330 million for imported packages).[165] This resulted in a net trade imbalance for software and computer services of over $850 million, and we estimate that this imbalance has grown annually by 20%-30% in the past four years.

7.9.3 Export Support

Canada's Export Development Corporation (EDC) provides financial services to exporters from Canada and offshore buyers. Instruments include a range of insurance, financing services and guarantees not provided by the private sector.

In terms of insurance, a Canadian firm of any size can insure export sales against non-payment by foreign concerns. Usually EDC assumes 90% of:

> "Commercial and political risks, involving insolvency or default by the buyer, as well as blockage of funds, war or cancellation of import licences and the like in a foreign country, and cancellation of export permits in Canada." (166)

EDC insurance pertains to both commercial and political risks involved in exports.

The export of goods, engineering services, software and technology may be insured, and a number of software firms have received insurance with respect to exports to the United States and Pacific Rim countries.

Insurance covers either sales of commodities and services made on short term credit loans of up to 180 days or goods and services made on credit up to five years. The following are the main EDC guarantee instruments:

1. Performance security guarantees, which cover financial institutions against any call of security (such as an international line of credit) which has been issued to a foreign purchaser on behalf of a specific Canadian exporter.

2. Bid security guarantees, which cover any bank which provides bid security to a foreign buyer on behalf of Canadian exporters.

3. Loan guarantees, which are issued to banks and financial institutions to provide loans to purchasers of Canadian capital goods and services.

4. Specific transaction guarantees, which provide banks and other financial lenders with "unconditional coverage on non-recourse supplier financing". The capital goods and services sale must be insured by EDC.[167]

The EDC also provides competitive export financing at both fixed and floating interest rates to foreign purchasers of Canadian goods, equipment and services. Funds are transferred directly from EDC to Canadian exporters on the borrower's behalf, thus providing the exporter with a cash sale. The following are EDC's main forms of financing:

1. A loan -- EDC loans pertain to single transactions.

2. Allocation under lines of credit -- A line of credit means a special kind of loan in which the foreign borrower (usually a banker or other financial institution) agrees to borrow monies from EDC for a multi-

plicity of transactions which neither the Canadian exporters nor the purchasers may have yet determined. "An allocation appears when a transaction, the buyer and the exporter have been identified and approved for financing under the line of credit. The line of credit may require several disbursement procedures, agreements, disbursement orders or some other form of disbursement arrangement."[168]

3. Multiple disbursement agreement loan -- In this type of financial agreement, the buyer (who is usually also the borrower) requires, for planning purposes, that specific financing terms be determined and set in the early stage. This form of loan is thought to have at least several exports involved in the project. Thus several disbursement procedure agreements are "linked" with the loan.

4. Note purchase -- EDC also purchases promissary notes which are issued by foreign buyers to Canadian exporters for purchases of Canadian goods or services. In this case, the exporter and EDC both sign a Note Purchase Agreement describing terms and conditions in the financing. Such notes may or may not be guaranteed by a bank.

5. Forfaiting -- This medium term financing service for small export sales of either goods or services involves a minimization of documentation and conditions involved in transactions. EDC purchases promissory notes which are issued by foreign buyers to the Canadian exporter for the purchase of Canadian goods or services. A prerequisite is that the notes be guaranteed by a bank which is acceptable to EDC. EDC purchases the notes from the exporter on a non-recourse basis at market prices.

6. Loan guarantees -- These guarantees are issued to financial institutions and banks providing loans to purchasers of Canadian goods and services.

The main types of exports which are covered on short term credit involve raw and semi-processed materials and goods, consumer goods and commodities, and engineering, technical and consulting services. EDC also insures against non-payment for "invisible exports" such as licensing or sale to foreign customers of rights involved in patents, trademarks or copyrights, leasing contracts, management fees, etc.

Although these export support instruments are generally now thought to be competitive with those of other countries, there is some evidence of regional favoring in the administration of these instruments.

An executive summary of an internal EDC report,[169] for example, has revealed that the lending activity of the EDC over the past three years has concentrated on Canadian exporters in Quebec. Also the report revealed that between 1981 and 1983, 53.9% of all EDC loans were made to five countries, with the US heading the list. (The other main EDC borrowers during the past three years included Algeria, Peru, Egypt and Mexico.) More significantly, during the past three years more than 50% of all monies lent by EDC supported sales of only ten companies, six from Quebec, and the most frequently user of EDC financing during this period was a crown corporation, De Havilland Aircraft of Canada Ltd., which received 29 loans during 1981-82 which were worth $340 million. During 1980-1982 an insignificant amount of available EDC monies (less than 1%) supported sales by software firms, and as Canadian firms enter the more risky software markets of the Pacific Rim, increased support will be necessitated, and regional imbalances will need to be addressed.

7.10 Export and Marketing Recommendations

Given the present small export penetration into two major opportunity areas of the world -- the US and Asia -- what types of measures might be instigated to better support software exports? Although export problems, we feel, are intricately tied to marketing and distribution problems, one obvious incentive would involve making it more profitable for software firms to export. However, most of the more aggressive export support measures, some of which have been enacted by other countries, are forbidden by GATT conventions, to which Canada is a signator.

Several countries, we have seen, have set up exemptions on imported technologies and materials used in the manufacturing of exports, and have reduced tax on all export revenues in selected industries. Several countries also set up preferential interest rates on loans for exports, according to which any exporter could borrow up to 90% of requirements at a reduced interest rate.

Other countries which have not signed the General Agreement on Tariffs and Trade have set up "indirect tax rebate certificates" for exports. According to these, an exporter of high technology goods and services can obtain a certificate of rebate of indirect taxes and utilize these certificates to pay this total tax rate, if the exports contain a national component greater than a certain percentage. (One country has used this certificate system to apply to exporters of technology and engineering/managerial services, and such certificates can be used to pay virtually any type of corporate taxes.)

Most of these export support measures are forbidden through GATT; however, it is clear that as software increases in economic importance, more violations will occur and countries will increasingly argue that software is exempted as a "special case" product. We thus recommend that the government closely monitor the introduction and effects of such export incentives by other countries as pertain to software and engineer/consulting services in general.

We also recommend that operating procedures of customs and excise with respect to intangibles such as software be reviewed and streamlined, and that intermediate inputs used in software production, such as AI workstations and other technologies, be freed of all tariffs and import duties.

We also recommend that the government be prepared to allocate a greater proportion of its aid-trade to software exports. This will be especially useful in the field of computer-based learning and training, since this field has evolved to the point where a large market is developing, and with resource software in Asian development projects. The latter may be closely tied to export of equipment, and (hence) may involve substantial dollar values that represent significant financial risk for exporters.

Although the history of high technology in developing countries is a disaster, and aid which is tied to sourcing from firms in donor countries has been repeatedly used to sell useless and inappropriate technologies, sometimes the development needs of Third World countries actually coincide with what industrialized nations have to offer. This situation is currently true for computer learning and training, still a very fluid market for virtually the entire world, and an area in which Canada will not be responding "five years after the facts" as in the increasingly saturated markets for office automation.

Beyond direct economic incentives for export, a number of governments have set up training institutes for software and other high-tech fields in attractive market areas of the world. These are not humanitarian gestures to aid developing countries but serve as an integral part of marketing, trade, investment and export promotion for the participating countries, since if one trains people in the use of specific technical services, one may create a future market dependence or preference for these.

In this work we have reviewed the success of the Singapore Institutes of Japan, France and Germany and noted the 1984 setting up of further Japanese software training institutes in China.

We recommend that the government investigate private sector interest and the costs and benefits in setting up a Canadian centre for educational/training technologies in the Pacific Rim region.

One location might be one of China's economic development zones. China will predominately benefit in the area of computer-based training by using it as a medium to acquire Western management and business know-how, a prerequisite for the day-to-day conducting of international business. (Here we are referring to matters as basic as accounting practices and computerized spread sheets.) Another major opportunity lies in the use of computer-learning to teach English. But even more, the Chinese need knowledge of, and facility in, using the new software development tools called applications generators in a wide variety of industries, so they can construct their own applications.

Singapore is another location offering several advantages. First is its centrality in Asia as both a communications and financial centre; secondly, Singapore has successfully implemented a strategy of converting itself into a centre of software expertise for Asia and the world and is a technology and software shopping centre for millions of Asian people.

In China, Singapore or some other Asian location, a Canadian centre could serve as a marketing conduit for Canadian educational technologies, software and services throughout the Pacific Rim. Canada's educational microcomputer, the Icon, for example, could serve as a vehicle for Canadian educational/training software throughout Asia. Possible direct long term benefits which could accrue to Canada include:

1. an improved and continuous access to market opportunities in Asian countries

2. greater export diversification and Asian investments

3. giving Canada an opening to a large emerging market in an area of the world where we enjoy a favourable ideological and political position

4. future sales of Canadian educational/training technologies and software arising from product commitment through training use.

Given the diffuse nature of the industrial benefits associated with a broad marketing instrument such as overseas software educational institutes, there is reason for concern that private sector participants may be reluctant to join in such a cooperative venture without government taking the initiative in this regard. Once established, however, it seems both appropriate and feasible for companies exhibiting in a centre to pay fees such that the centres create no significant need for continued government subsidy.

Turning now to marketing, costs for packaged software, we have seen, are substantial. However, with promising offshore markets 17 hours away in the

Pacific Rim, travel costs incurred in marketing custom software may themselves be significant.

There is much more risk involved in the development of packaged than custom software, since one receives periodic funds for making the latter, while with the former the package developer usually pays all costs from internally generated funds, and cash flows only begin after the product has been marketed and sold.

The most common way Canadian software firms have entered export markets in the US and Asia is to set up joint ventures, usually licensing and distribution agreements with local vendors, since there are prohibitive costs involved in setting up marketing and distribution operations offshore.

Although tariff and non-tariff barriers to software trade between the US and Canada are presently minimal, continuing access to the US markets is crucial for industrial growth. (We need access to their software market more than they need access to our market for computers.) Relevant to this fact, an Ontario software study[170] has recommended that the federal government negotiate a North American Free Trade Area (NAFTA) with the US for computers and software and that Ontario set up a software marketing board. Although the functions and effects of this board are unclear, the main reason for a sectoral free trade agreement with the US is to guarantee that products of Canadian firms will continue to have marketing and distribution outlets in the US. Unfortunately this area is one of great controversy for Canada and immediately involves issues of transborder data flows, cultural sovereignty, broadcast policy, etc. But if a central issue in software marketing involves control of and access to the distribution channels, it is important that we do not repeat the same errors involved in protective policies toward software industries as has been the case with Canadian film and television industries. In principle then, we agree with the notion of an informatics free trade area, recognizing the fact that this issue is fraught with controversy.

On a more practical level, since Canada's economic development will increasingly depend on knowledge-intensive industries such as software, and since many software firms lack knowledge of export opportunities, we recommend that funding for the Program for Export Market Development be increased, with special emphasis on the United States and Pacific Rim countries.

We also recommend that the Department of External Affairs, through its trade commissioners, in conjunction with the several regional software development associations, set up a program designed to get off-shore market information to Canadian software suppliers -- including information about buyers, financial intermediaries, market trends, relevant development projects off-shore, etc.

The government cannot, and should not, attempt to do marketing for the private sector, nor is it likely that government programs are going to improve the marketing competence of firms and individuals. But the government can certainly facilitate marketing through a wide variety of programs such as aid-trade in the bilateral programs of the Canadian International Development Agency (CIDA) and through improving market information, where that information is of a broad "public-goods" nature, e.g. foreign government initiatives affecting markets for Canadian software products.

Throughout Canada regional software industry developments associations are forming, and each of these associations are contemplating forming software marketing groups, whose purpose, through a variety of opportunities, involves opening windows for access to software publishers and distributors, facilitating joint ventures, trade shows and related marketing channels. We feel that industry spokespersons of these regional software associations are better suited than government to direct marketing activities. Such marketing associations will also provide advice and assistance concerning market research, the preparation of business and marketing plans, legal patent and copyright aid, and advice on raising capital, and their involvement in government marketing programs is crucial.

In conclusion, as our discussion of market failure suggested, there is no compelling theoretical reason (beyond conventional risk aversion) for why private investors should be unwilling to provide support for proprietary software products that are technically successful and that have potential market applications. There is no doubt that investors in software companies would like to pass some, or all, of their necessary marketing costs onto public taxpayers. But unless social efficiency is improved, as a result, there is no public interest in doing so.

In practice, one can appeal to an argument that Canadians are "excessively" risk-averse as private investors, and that this is why many prototypes are developed in Canada but are produced elsewhere in the world. That is, funds cannot be generated at a reasonable cost domestically. Hence, Canadian entrepreneurs must sell their proprietary technologies to foreigners for commercial development.

There is ample anecdotal evidence to support this theme, and it is clear that marketing costs are a substantial portion of the overall costs of commercializing software. To this extent, we feel it is appropriate for policymakers to consider implementing a program whereby long term loans are made available to software producers for undertaking market development and distribution activities. The loans should be at rates competitive with those paid by larger "higher technology" companies.

7.11 Concluding Remarks

The heart of our recommendations involves tax/fiscal policy. Taken as a whole, they argue for increased (primarily indirect) subsidies for the Canadian software industry. While such increased subsidies can be theoretically defended on standard grounds of economic efficiency, we must also acknowledge that market failure-based arguments for government intervention offer no clear guide to the "optimal" magnitude and extent of public sector intervention. Nor do the arguments exclude the possibility that the design and administration of specific programs will fail to elicit the desired response. Indeed, the possibility exists that any program of subsidies will merely result in a transfer of wealth from taxpayers to corporate shareholders and/or to specific scarce factors of production.

In this regard, a recent survey of 55 firms (which account for almost 30% of the company-financed R&D expenditures in Canada) indicated that in 1982 the investment tax credit increased company-financed R&D expenditures by about 2%, and that the special research allowance increased them by about 1%. However, the increases in company-financed R&D expenditures due to these tax incentives seem to have been considerably less than the cost to the government in reduced tax revenue. This evidence for Canada indicating that tax incentives have had only a modest effect on R&D spending is similar to results for Sweden and the United States.[171] All of this is by way of cautioning that all tax/fiscal programs should thus have "sunset" features which require re-examination of the effects of the programs before renewal is extended, but the term of the programs should be sufficiently long to facilitate planning.

REFERENCES

1. John Evans, "An Economic Analysis of the Canadian Software Industry," Department of Comunications, Ottawa, 1983. Recent provincial studies include Touche Ross and Partners, "Study of the British Columbia Software Industry," Vancouver, March 1984, and Zavich Zeman et al., "Ontario Software Industry," ZZ International, Toronto, February 1984.

2. Cognos, "The State of the Art in Natural Language Automated Processing and Artificial Intelligence," "Implications of New Technology Thrusts in Natural Language Processing and Artificial Intelligence," "The Social and Economic Impact of Possible Technology Developments in Natural Language Automated Processing and Artificial Intelligence," "Opportunities for Canada in Natural Language Automated Processing and Artificial Intelligence," "Implementation Plan for Natural Language Automated Processing and Artificial Intelligence," "Priorities for Canada in Natural Language Automated Processing and Artificial Intelligence," Secretary of State and Department of Communications, Ottawa, 1984.

3. Business Week, February 27, 1984, p.91.

4. Ibid., p.92.

5. EDP In-Depth Reports, April 1983, p.6.

6. Software Canada, June 1983, p.10.

7. Judith Drake, "Software Manufacturing: A Case Study," cited in On-Line Data Access, 1984.

8. D.D. Cowen and J.W. Graham, "A Study of the Packaged Software Industry in Canada," DRIE, Ottawa, February 1983, p.11.

9. Remarks at 1983 Washington, DC Conference on Artificial Intelligence.

10. See, for example, J. Hirchleifer, "The Private and Social Value of Information and the Reward to Inventive Activity," American Economic Review, September 1971.

11. Business Week, February 27, 1984, p.96.

12. EDP In-Depth Reports, May 1982, p.4.

13. Lientz and Swanson, "Software Maintenance -- The User Management Tug-of-War," Data Management, April 1979.

14. EDP In-Depth Reports, May 1982, p.3.

15. Ibid., p.6.

16. Computer Communications Secretariat, "Trans-Border Data Flows - Software Related Issues," Department of Communications, 1981.

17. Beau Sheil, "Power Tools for Programmers," Datamation, February 1983, p.134.

18. Ibid.

19. Ibid., p.138.

20. The Economist, January 9, 1982, p.81.

21. Datamation, August 1983, p.95.

22. Datamation, October 1983, p.92.

23. Ibid., p.93.

24. Cognos, "Opportunities for Canada ..."

25. Ibid.

26. Cognos, "The State of the Art ...", pp.15-16.

27. Ibid.

28. Ibid.

29. Ibid.

30. Fortune, March 5, 1984, p.88.

31. The Economist, May 28, 1983.

32. The Economist, February 26, 1984.

33. Business Week, January 30, 1984, p.50.

34. Quoted in Business Week, October 31, 1983, p.95.

35. Russel M. Wills, "Microprocessor-Based Media," IDRC, Ottawa, 1981, p.15.

36. The Economist, April 24, 1982, p.114.

37. Juan Rada, "Structure and Behaviour of the International Semi-Conductor Industry," UNCTC, New York, 1982, p.212.

38. Russel M. Wills, "Softech: A Microelectronics Guide," Ministry of Industry and Trade, Queens Park, Toronto, 1982.

39. Financial Times of London, July 2, 1979, p.5.

40. OECD, "Software: A New Industry," Paris, February 1984, p.62.

41. Datamation, October 1983.

42. OECD, op. cit., p.15.

43. Business Week, April 18, 1983, p.92.

44. Business Week, February 27, 1984, p.75.

45. Business Week, February 27, 1984, p.75.

46. Business Week, February 27, 1984, p.82.

47. OECD, op. cit., p.52.

48. Ibid., p.97.

49. Business Week, November 21, 1984, p.114.

50. Frost and Sullivan, 1984.

51. The Economist, January 21, 1984, p.79.

52. Business Week, February 21, 1983, p.18.

53. Business Week, August 29, 1983.

54. Business Week, June 27, 1983.

55. Wills, "Microprocessor-Based Media," p.141.

56. J.E. Crozier, "A Study to Identify the Attitudes and Awareness of the Machine Control Users to CAD/CAM Technology," IT&C, November 1980.

57. MMBTA Survey, 1982.

58. EIA Standards Proposal, 1546A.

59. The Yankee Group, 1983.

60. OECD, op. cit., p.138.

61. EDP In-Depth Reports, May 1983, p.2.

62. Allen Kay and Adele Goldburg, "Programming Your Own Computer Personal Dynamic Models," Computer, March 1977.

63. Byte, August 1981, Special Issue on Smalltalk.

64. Robert W. Lawler, "Some Powerful Ideas," MIT AI Lab (AI Memo 652), December 1981.

65. J.W. Brahan and W.D. Godfrey, "A Marriage of Convenience: Videotex and Computer-Aided Learning," Computers and Education, 6: pp.33-38.

66. W. Olivier, "CAN-8: A Complete Instructional System for Alphanumeric Graphic Terminals or Optical Videodisc," Proceedings of the Third Canadian Symposium on Instructional Technology, Ottawa, NRC, 1980.

67. M. Westrom et al., "The B.C. Tel Experience," CIP Review, November/ December 1981, p.27.

68. Ibid.

69. J.W. Brahan, "NATAL-74: Concept to Reality," unpublished paper.

70. Byte, op. cit.

71. S. D'Antoni and J.W. Sprecher, "Computer-Assisted Instruction: Current Trends and Critical Issues," Communication of the ACM, 23, June 1980, pp.332-342.

72. Kalba Bowen Associates Ltd., "The Expanding Home Video Market," Cambridge, Mass., January 1981.

73. Statistics Canada, "Statistics in Education, 1981-1982." This amount includes all expenditures on universities, community colleges and elementary and secondary schools.

74. The Computer Teacher, October 1983.

75. In 1979, Canadian publishers reported a net sales of $91 million for elementary/secondary school texts out of total sales, including imports of $132 million. Textbook sales are thus the second largest category of book sales after general readership books (private communications and Statistics Canada).

76. This is further complicated by the fact that the government, through special programs, pays a portion of employee's wages during training, but this portion varies from program to program.

77. Globe and Mail, October 12, 1981.

78. Private communication, B.C. Tel Education Centre.

79. Globe and Mail, December 8, 1982.

80. Beth Brown, "Characteristics of Industry-Based Programs: Selected Case Studies," Office of Technology Assessment, Washington, DC, July 1981 (draft).

81. National Research Council, "Proceedings of the Third Canadian Symposium on Instructional Technology," held in Vancouver, BC, February 1980.

82. HUMMRO, personal communication.

83. Ibid.

84. John Evans, "An Overview of the Canadian Software Industry," Department of Communications, August 1983.

85. Cognos, op. cit., p.40.

86. EDP In-Depth Reports, September 1983.

87. Nigel Kelly, Software Canada, August 1983, p.6.

88. EDP In-Depth Reports, October 1983.

89. Juan Rada, op. cit.

90. H. de Mattas, ITV Telecommunications Forum, 1979, Vol. I, p.II 2.5.

91. SEI, Brazilia, 1982.

92. SEI, "Transborder Data Flows," 1982.

93. SEI, personal communication.

94. SEI, Brazilia, 1982.

95. Ibid.

96. Special Commission on Software, SEI, Brazilia, 1982.

97. National Computer Board of Singapore, personal communication.

98. Annual Report, Economic Development Board (EDB) of Singapore, 1983.

99. National Computer Board Publication ISSN-0217-6432, January 1983.

100. Asza, personal communication.

101. National Computer Board of Singapore, personal communication.

102. Industrial Development Department, National Computer Board, "Singapore Computer Industry Survey," Singapore, October 1983.

103. Annual Report, EDB, 1983, p.37.

104. National Computer Board, personal communication.

105. Annual Report, EDB, 1983.

106. Lityan Group, personal communication.

107. Business Week, August 5, 1983, p.9.

108. Ibid., p.95.

109. Business Week, July 11, 1983, p.71.

110. Ibid.

111. Computerworld, December 29, 1983, p.65.

112. Cognos, "The State of the Art ...", p.149.

113. J. Alvey et al., "Programme for Advanced Information Technology," HMSO, 1982.

114. Business Week, March 19, 1984, p.71.

115. STI Indicators, Tokyo, 1981, pp.22-23.

116. Business Week, May 2, 1983, p.52.

117. Business Week, February 27, 1984, p.95.

118. Fortune, December 12, 1983, p.98.

119. Ibid.

120. Business Week, January 17, 1984, p.128.

121. The Economist, November 26, 1983, p.71.

122. Business Week, January 10, 1983, p.96.

123. Business Week, May 23, 1983, p.87.

124. US Census Bureau, Washington, DC, 1982.

125. Economic Council of Canada, "Intervention and Efficiency," Ottawa, 1982, p.27.

126. Ibid., p.27.

127. Ibid., p.28.

128. J. Panzar and Robert Willig, "Free Entry and the Sustainability of Natural Monopoly," Journal of Public Economics, August 1972.

129. Touche Ross and Partners, op. cit.

130. Business Week, "Software: The New Driving Force," February 27, 1983.

131. S. Globerman, "The Relationship Between Formal Education and the Adaptability of the Workforce to Technological Change," Study for the Royal Commission on Economic Union and the Development Prospects for Canada, 1984.

132. J.S. Cohen et al., "Chasing the Bandwagon: Government Policy for the Electronics Industry," Canadian Public Policy, Vol. X, 1984, pp.25-34.

133. Statistics Canada and DRIE estimates.

134. Report of the Ontario Microelectronics Task Force, Queens Park, 1981.

135. D.P. Demelto et al., "Innovation and Technological Change in Five Canadian Manufacturing Industries," Economic Council of Canada, October 1980.

136. Cognos, see reference 1.

137. Ibid., p.26.

138. Ibid., p.138.

139. Ibid., p.139.

140. Globe and Mail, Friday, March 16, 1984, p.B15.

141. Ibid.

142. EEMAC, "Maintaining the R&D Incentive," Toronto, February 1984, pp.3-4.

143. MOSST, Directory of Scientific and Technological Capabilities in Canadian Industry, Ottawa, 1977.

144. EEMAC, op. cit.

145. "Draft Amendments to the Income Tax Act with Respect to Research and Development," Department of Finance, Ottawa, 1983.

146. Winnipeg Speech, 1983.

147. The Financial Times, March 31, 1984.

148. The Globe and Mail, Monday, June 4, 1984, p.12, and personal communication, Esso.

149. The Globe and Mail, June 11, 1984, p.14.

150. Ventures West Technology, personal communication.

151. Peter M. Farwell, "Taxation of Computer Software Costs," Clarkson Gordon, Toronto, undated.

152. Ibid., p.20.

153. See Jeffrey Bernstein, Research and Development: Production, Financing and Taxation, Ontario Economic Council, University of Toronto Press, 1984.

154. Business Week, April 30, 1984, p.108.

155. Computerworld, February 6, 1984, p.26.

156. See, for example, Parker vs Floak, 1978.

157. R.M. Miligrin, "Program for Lawyers Doubly Deterred," Datamation, March 1983, p.160.

158. Ibid.

159. N. Bryce and Associates vs Gladstone, 1982.

160. J. Palmer and R. Resendes, "Copyright and the Computer," CCA, Ottawa, undated, p.20. See also the federal white paper on copyright revision, "From Gutenburg to Telidon," CCA, 1984.

161. Wescom estimates.

162. Teresa Plowright and Russel Wills, Computer as Tutor: The Policy Environment for Computer Learning in Canada, IRPP, Toronto, in preparation.

163. NRC, "Towards a National Policy for a Computer-Assisted Learning Industry," Ottawa, 1984.

164. Business Week, May 23, 1983, p.77.

165. DRIE estimates, 1984.

166. Export Development Corporation, publication 80-1.

167. Ibid.

168. Ibid.

169. Reported in the Globe and Mail, August 7, 1984, p.B1.

170. Zeman et al., op. cit.

171. Relevant studies are discussed in Edwin Mansfield, "Technological Change: A Survey of Findings," University of Pennsylvania, Mimeo, 1984. See also D. McFetridge and J. Warda, "Canada R&D Incentives: Their Adequacy and Impact," Toronto: Canadian Tax Foundation, February 1983.