OCCASION

This publication has been made available to the public on the occasion of the 50th anniversary of the United Nations Industrial Development Organisation.

DISCLAIMER

FAIR USE POLICY

CONTACT

Please contact publications@unido.org for further information concerning UNIDO publications.

For more information about UNIDO, please visit us at www.unido.org

UNITED NATIONS INDUSTRIAL DEVELOPMENT ORGANIZATION
Vienna International Centre, P.O. Box 300, 1400 Vienna, Austria
Tel: (+43-1) 26026-0 · www.unido.org · unido@unido.org

INDUSTRY AND DEVELOPMENT

No. 28

UNIDO

UNITED NATIONS
INDUSTRIAL DEVELOPMENT ORGANIZATION

Vienna, 1990

AIMS AND SCOPE OF *INDUSTRY AND DEVELOPMENT*

Industry and Development attempts to provide a link between practitioners and theorists working on economic and related aspects of industrialization. The focus of the journal is on applied economics, particularly in areas emphasized in the Lima Declaration and Plan of Action on Industrial Development and Co-operation.

The journal is published an average of four times a year as an integral part of the work programme of the Studies and Research Division of the United Nations Industrial Development Organization. It is prepared under the general guidance of a Supervisory Panel, composed of staff members from the Division, with the Head of the Global Studies Branch as its chairman. Responsibility for the detailed supervision of a specific issue is rotated among the members of the Panel. The responsible member for this issue was J. Cody.

The Supervisory Panel of *Industry and Development* welcomes readers' opinions and comments, and will be glad to consider for possible publication articles relevant to the aims and scope of the journal (see "information for contributors").

ID/SER.M/28

UNIDO PUBLICATION
Sales No.: E.90.III.E.10
ISBN 92-1-106248-9
ISSN 0250-7935
02400P

Explanatory notes

References to dollars ($) are to United States dollars, unless otherwise stated.

The term "billion" signifies a thousand million.

In tables:

Totals may not add precisely because of rounding.

A hyphen indicates that the item is not applicable.

An em dash (—) indicates that the amount is nil or negligible.

The following abbreviations are used in this publication:

AI	artificial intelligence
ANSI	American National Standards Institute
CAD	computer-aided design
CAE	common applications environment
CAM	computer-aided manufacturing
CIM	computer-integrated manufacturing
ECC	European Economic Community
GNP	gross national product
IPSE	integrated projects support environment
ISDN	integrated services digital network
ISIC	International Standard Industrial Classification of all Economic Activities
ISO	International Standards Organization
NIC	newly industrializing country or area
OECD	Organisation for Economic Co-operation and Development
OSI	Open System Interconnect
SITC	Standard International Trade Classification
WIPO	World Intellectual Property Organization

18254

THE SOFTWARE INDUSTRY: DEVELOPING COUNTRIES AND THE WORLD MARKET

Eoin Gahan*

Introduction

This study analyses the software industry from the point of view of a new entrant to what is now a complex and world-scale market. A new software producer faces new opportunities and difficulties, and to be successful requires a wide variety of skills.

Two main trends dominate the industry. The first is the growth of software as a traded product, rather than an individual service. Software is designed, produced, packaged and marketed on a large scale and in a sophisticated way.

The second trend is the influence on software of technological change in hardware. In general, hardware developments have moved ahead of software's ability to exploit them fully. The variety and pace of hardware change means that many opportunities exist for new software products.

UNIDO has for some time been emphasizing the importance of the software industry for developing countries, and the need to incorporate an awareness of its significance in national technology policies and programmes. The Technology Programme of UNIDO has produced a number of documents dealing with several technical aspects of software and its production, and detailed legal and institutional questions have also been a particular focus.**

A. Software: its nature and significance

1. Definitions of software

Software is a set of instructions to a computer. It is thus distinguished from, for instance, hardware, which is any component of the computer itself, or data, which is what the computer uses or generates to make output, which itself is the reason why the

---

*Secretariat of UNIDO, Regional and Country Studies Branch.

**See, for instance, "Trends in commercialization of software in developing countries" (UNIDO/IS.574), "Guidelines for software production in developing countries" (UNIDO/IS.440), and, most recently, "Software production: organization and modalities" (UNIDO/IPCT.63). A regular coverage of software is also found in the UNIDO Micro-electronics Monitor.

software is supplied to the computer. This definition, though not a rigorous one, has the major advantage that it focuses on the use of computers as a process, and makes a distinction between the ultimate product and the means of transformation. Many alternative definitions exist such as that "hardware is what you can touch, and software is what you can't touch". This however does not make clear the distinction between software and data.*

The Organisation for Economic Co-operation and Development (OECD), in its first study on software, adopted a definition referring to "... a combination of data and instructions, many being algorithms ..." ([1], p. 20). In the present study, however, a narrower definition is preferred both because it delimits to some extent what is still a very large topic and also because of the quite different market forces at work.

The same OECD study also provides definitions from the World Intellectual Property Organization (WIPO) and from the International Standards Organization (ISO). The WIPO definition appears to be rather broad, since it includes not only computer programmes but also descriptions of them and instructions for their use. The ISO definition also includes documentation, and contains the important additional statement that "software is independent of its carrier media".

At first sight the last statement appears unexceptionable, and seems to be in conformity with the "intangibility" quality mentioned above. But it raises some difficulties in the detailed consideration of software and especially its commercial aspects. In practice, software is not independent of its "carrier media", and the disembodied view of software is inadequate for an understanding of the ways in which developments in software are related to those in hardware, and of the ways in which software is marketed.

2. The significance of software

Whatever precise definition is used, the idea of a set of instructions is almost always understood as the underlying nature of software. Instructions are commands to the hardware, and at a fundamental level it can be seen that the hardware determines what software is written. This is because the set of all possible instructions is determined by the hardware: there is no point in supplying an instruction that cannot be acted upon by the machine which receives it.

The basic instructions which the machine understands (and is built to understand) are called the "machine language". They can usually be represented by binary digits, that is, a string of 1s and 0s, or perhaps by hexadecimal codes (numbers to the base 16).

*At the level of machine language, however, this distinction can become rather tenuous.

In any case, manipulation of these codes is usually simplified by a piece of software which allows the programmes to use mnemonics such as ADD or STORE rather than a string of difficult-to-recognize digits. Such a programme, usually called an assembler, would in turn be used to write another programme which would allow more powerful commands, and so permit the instructions for mathematical or other manipulations of data to be expressed in a form nearer to the formal way in which such manipulations would be described in speech or in writing.

The set of such formal instructions is usually called a high-level language. Examples include COBOL, FORTRAN, C and Pascal, and they are discussed in more detail in section D of this study. They are mentioned here only to show the way in which software in one sense is indeed independent of the hardware. A definition of C, for instance, usually says nothing about the type of computer on which it can be used. In general, C as a concept is hardware-independent, just as most other high-level languages are also.

However, for a computer programmer actually to use C on a computer, in the sense of running a programme written in C and obtaining results from the data being analysed, there has to be a means of converting C instructions (usually called statements) into the machine language of the computer concerned. As noted above, this means of conversion is in fact another piece of software, and it is usually called a compiler, since it takes all the C statements and converts them to a set of machine commands. Another sort of translator is called an interpreter: it converts line by line. Unless a compiler or interpreter exists, C cannot be used on the computer in question. Thus, although in theory a piece of software may be machine-independent, it is not really so in practice.

There is a further problem which occurs not only with software but also with data. This relates to the reading by the computer of software and data, usually in the form of magnetic media such as diskettes and tapes. These themselves are hardware, but the way in which the software and data are stored on them varies much more than the hardware does. Thus a piece of flexible plastic in the form of a disk, wrapped in a flexible cover, may be identical for many different microcomputers, all of which are capable of understanding the software written on it. And yet each of them may use a different code to represent the characters and, more probably, each will also use a different way of laying out the data and software on the disk. Some might use one number of divisions, some another. Some might use concentric ring patterns and some a spiral pattern.*

---

*This is in fact the case with the IBM Personal Computer System 2 and Apple Macintosh computers, both of which use the same size diskette (3.5 inch, or 8.9 cm).

The role of software is not only in computers *per se*, even though these are being used as examples in the present discussion. Software is the set of instructions stored in anything that is programmable, and this includes many different kinds of machinery and equipment, from machine tools to telephone and communications equipment and dispensing and vending machines. Increasingly such machines embody software to control their operations, not only because it is much easier and cheaper to solve many problems in terms of programmed micro-electronic circuitry, but also because it makes it much easier to make changes in the way the machine operates and thus to adapt it to different uses.

The case of numerically-controlled machine tools is a particularly important one. It is the spread of micro-electronics which has allowed these to become even more flexible and powerful, even though the basic ideas go back to the punch cards used to change control of weaving equipment in the textile mills of the nineteenth century. The programmable character of modern machinery means that it can more easily be adapted to changes in the design or the content of the product being produced, thus allowing for very rapid retooling in the face of changes in demand patterns, competitive pressures or changes in primary input costs. Software thus is a usually hidden but essential element in many aspects of manufacturing. The spread of robotization, factory automation, computer-aided design (CAD), computer-aided manufacturing (CAM) and computer-integrated manufacturing (CIM) means that software will have an even more important role in the future, especially in the linking of the software embodied in individual machines into a fully communicating and controlled system of production.

It is, however, software for computers which is most easily discussed, especially because here is most easily explained the concept of software as a traded commodity.

3. Software as a commodity

The International Standard Industrial Classification (ISIC) of the United Nations is intended to cover all economic activities, and being based on a structure first conceived in 1948, does not cover the production of software anywhere. Equally, the United Nations Standard Industrial Trade Classification (SITC), Revision 2, does not deal with software at all. This is more surprising because it dates from 1975 ([2], [3]). The exclusion of software from classification systems may be partly because of its (relative) newness, but a major reason must also be its intangibility, as referred to earlier.* The easy option may have been taken of treating the software industry as a service with no permanent tradeable outputs. This approach may explain its exclusion from SITC, and its brief history may be the reason why it is not found in ISIC.

---

*Software is sometimes specifically referred to as an "intangible (fixed) asset" in taxation regulations. See [4].

However, the fact is that software is now an industry of global reach, whose products are conceived, designed and marketed in a typical manufacturing-type operation. The writing of software to order still continues extensively and will continue for a long time to come.* But increasingly it is the mass production of software as a tradeable good which represents the visible shape of the industry.

Complex issues arise, not just at the level of statistical classification, but at the very practical levels of customs tariffs, cross-border data flows, patents and copyright law. More fundamentally, they arise in the questions of the nature and future shape of the industry, the evolution of the manufacturing process and the internationalization of production. So rapidly is the role and nature of software developing that a fully comprehensive view of its scope is not easy. However, certain characteristics are visible which suggest that it can usefully be regarded as an industry producing merchandise which is traded. This aspect of software is growing rapidly. The industry is labour-intensive, but also skill-intensive. In addition, technological change affects the industry, with a tendency towards increased capital inputs, although clearly the overwhelming character of the industry is that of a (skilled) labour-intensive one. Technological change has also affected product life cycles, which are continuing to shorten, but perhaps the biggest influence on this tendency has been the increasingly fierce competition among producers. A search for new product ideas is a continuing necessity for survival.

With respect to technological change, the time needed for most innovations is substantial. This is because of the labourand skill-intensity of software development. Thus, between the conceptual definitional stage of the new software and its fault-free, saleable form, a considerable time, sometimes years, may be needed. New tools may allow for a shortening of development time, both by increasing programmer productivity and by simplifying the complex managerial tasks associated with large software projects. However, one interesting effect of the long product development times for new technological levels of software is that the future shape of software can be predicted with more confidence then for other products of industry. Thus, areas of search for new opportunities can be more easily identified. In general, the software industry, although many large firms are found, remains open to newcomers. It offers potentially large rewards for innovation. Because of constant change in technology, it provides many niches for which distinctive products can be developed.

---

*But not necessarily in familiar ways: the construction of "macros" in spreadsheet applications, and even the storing of key sequences in a function key is a kind of programming and thus amounts to software creation.

4. Software and developing countries

For developing countries such considerations are particularly important. Software skills are in any case necessary to maintain or improve the competitiveness and efficiency of all sectors of their economies, not just manufacturing. However, only mastery of such skills can allow a developing country to control the direction of its informatics development as a whole, since reliance on imported software will direct its production structures into ways of doing things which are determined elsewhere. This can be a difficulty not only from a cultural point of view. It can raise problems also in very practical ways. Imported software may require data in a form in which it is not usually kept in the country, or may assume a particular system of accounting or taxation which is usual in developed countries.* To use them may mean a complete alteration of patterns of work and ways of thinking. Often the resulting confusion could wipe out any gains in efficiency made by computerization. Even if this is not the case, there is still the problem of interfacing with other, manual systems which use the accepted national approach to the problem in question.

The mastery of software skills becomes especially critical if the informatics field is recognized as a main means by which developing countries can increase their international competiveness. The developing countries' cost advantages of low wages can be and in fact are being quickly eroded by heavy investment in automation and CIM in developed countries. Developing countries in any case face enormous problems of co-ordination in enabling their economic and social systems to improve the living standards of their people, and informatics technologies can contribute significantly to their solution.

These arguments for informatics development in developing countries are well known. They are restated here because an additional point is to be made, which is that, in evaluating the scope of software development in a developing country, the external market has to be considered.

The export of software, which can be to another developing region or the world market as a whole, has distinct advantages for developing countries which require that it be seriously considered as a policy option. First, it should be recognized that software is an industry like any other with advantages (and disadvantages) as a component of an export strategy. But secondly, there is the vital point that a concentration on exporting software will promote quality and technological progress in the domestic industry in general, in a way that concentration on, for instance, clothing manufacture or food processing will not. This is because

*Schwarz [5] cites the examples of the Burmese Post and Telecommunications Corporation, which acquired payroll software that allowed for far fewer standard deductions than are usual in the Union of Myanmar (formerly Burma).

competition in the latter field is based in so many cases on price alone, and in software exports this can never be the case.

The improved quality of a software industry oriented to the external market means in turn that the rest of the developing economy concerned then has direct access to strategically important components for improving organizational efficiency, increased productivity, enhanced competitiveness and the flexibility needed to respond to rapid changes in economic conditions. The increased application of micro-electronics, computers, telecommunications and automation all require a mastery of software skills. A competitive software industry in a developing country, which succeeds in selling software products in a very challenging and rapidly shifting world market, is a national asset which can contribute to most other economic activities within the country.

B. Overview of the industry

1. Types of software

In this section, a statement of the kinds and classes of software that are marketed is attempted, not as a basis for a definitive taxonomy, but in order to provide the working language needed for a discussion of the actors in the software industry, the roles they play and the strategies they follow.

As noted above, the working definition of software is restricted in this study to include only instructions to the computer or other programmable device. This, in fact, could make the definition of software the same as that of programmes, but it is probably better to regard software as a generic term, covering all traded software. The units of software may be a programme or a set of programmes (sometimes called a "suite") if they are related to one another.

Software can be written by the user ("developed in-house") or it can be acquired from an external supplier. Sometimes this doesn't cost anything. That can be the case if the software is illegally copied. But it can also be the case if the software comes from a university, for instance, or a voluntary group of users of a particular type of computer, or a professional society. This is called public-domain software and is particularly important for microcomputer software (such software can often be worth investigating, before embarking on a major software development project).

In turn, the software acquired externally can either be written to order to meet an individual requirement ("custom software") or it can be already written and on general offer. This is called packaged software.

The main types of software are systems software and applications software. Systems software is the software that helps the computer to work, and applications software is the software that does the work which makes the computer useful. Thus, the payroll programme, which calculates the monthly salaries to be paid, is an applications programme, while a disk utility programme, which organizes the various files of information in some way on a magnetic storage device, is an example of systems software.

Systems software includes, most importantly, the operating system of the computer. This is the software which governs the interaction of all the components, interrogating the operator's keyboard and writing to the screen scheduling tasks, deciding on priorities, organizing storage etc. Operating systems can be very rudimentary or extremely sophisticated. They are crucial in that they usually determine the environment for any other software to be used on the computer. An operating system is almost always present, and any other software to be used on the computer has to be compatible with it. Thus, to take an example, MS-DOS is an operating system for computers such as the IBM personal computer, and the software to be used with it is said to "run under MS-DOS". If an operating system is sufficiently sophisticated, there may be no other systems software available or needed for the computer, but more usually there will be so-called utilities to improve upon the tasks carried out by the operating system or to supplement them. Another example of systems software is a special programme to handle communication with computer terminals or with other computers or devices.

Application programmes include packages for automating the calculations involved in almost every field of human activity. A package is a term applied to a more or less stand-alone computer programme which offers full control of input and output and of storage within the programme. It will typically be able to carry out a wide variety of different tasks within a subject area. No hard-and-fast rules separating programmes from packages have been drawn up, but the ability to issue alternative commands to a programme may be the key characteristic that determines whether it is a package or not. Thus, a spelling checker is probably a programme: it takes a file of text and compares each word against a dictionary to find errors. But word-processing software is a package: there is usually a sufficient number of choices of things to do, such as editing, merging, cutting, pasting, searching and word-counting (a spelling-checker will probably also be incorporated).

As noted above, "packaged software" is something rather different. Here the word package is used to emphasize that what is being sold is being treated as a consumer good, a product which could be found on shelves in shops. The software, usually on diskette, with a user's manual, wrapped in a box, is marketed often with the same skills as are used to market any other fast-moving consumer good.

The distinction between systems software and applications software is not always absolute. A data base package, for instance, is systems software in that it is concerned with the organization of data and its storage, and it may arrange all the data on the disks of the computer, superseding the existing arrangement followed by the operating system. Similarly, it may carry out communications and other functions normally carried out by the operating system. But a data base programme, from another point of view, is applications software, because it usually provides tools for sorting, selecting, totalling, and, in general, what is called report generation, that is, the carrying-out, as required, of specific analyses of the data in the data base and the presentation of the results in an acceptable form. Computer language software is also difficult to classify; arguments for both points of view could be found.

But the distinction between systems and applications is still a useful one, and will be followed in this paper. There is plenty of software which is unambiguously of one kind or the other, and the classification helps to analyse the industry in more detail and its relation to the hardware question. The two types of software are increasingly being sold to distinct markets. Systems software is largely bought by computer specialists and professionals, and applications software more by non-specialists who are interested in the computer only as a tool to assist in accounts, engineering, medicine, law, customs clearance or some other activity. It has been increasingly noted that even though the actual purchases of applications software may be made, in a company, by the data-processing department (that is, the traditional computer users), the choice may be as a result of pressure from the non-computer staff who have identified the piece of software as useful for their own work. The so-called end-users thus exercise increasing influence, and marketing efforts are increasingly directed towards them.

2. Software producers

The main types of producers are as follows: computer manufacturers; software companies; original equipment manufacturers; value added resellers; system houses; and computer users.

The computer manufacturers are major producers of software. In the past, computer manufacturers used to provide almost all the software that was available for the computer in question, apart from that written by the purchaser. There was thus no so-called third-party software available. Manufacturers have to ensure that there is enough software available to encourage purchase of the machine, and this they do either by following an existing hardware design, or by providing a version of a standard operating system, for both of which sufficient software is considered to be available.

Software companies are difficult to classify. They range from very small enterprises, with perhaps just the proprietor writing and selling the software, to transnational corporations with complete international distribution systems and full service and support networks. The common characteristic of software companies is the high ratio of skilled employment to fixed assets. Certainly the company will have computers on which to develop the software, but the main assets are its people. The research and development phase is the crucial and the expensive one: production costs have been low up to now, involving merely the duplication of a tape or diskette. However, increased competition has led to a more demanding market for documentation. The companies have to provide detailed manuals and usually spend a good deal in making them attractive. The need to prepare different language versions is a further cost of production. In addition, sales and marketing expenditures in a highly competitive field have become increasingly significant. Also, apart from documentation per se, the purchasers of the software will expect, depending on its price, a range of services to be provided by the software producer. This includes telephone support (such as a "toll-free hot line") which provides the user with help and advice. Other services provided can include newsletters, user groups etc. While the user may be charged a fee for some of these, there is usually a subsidy from the software producer also.

The name "original equipment manufacturer" is misleading, since what such manufacturers do is to put together a package of hardware and software, put their own name on it and sell it. Thus, they might buy a basic computer from one manufacturer, memory and peripherals from another, and software from a third. The original equipment manufacturer may, however, also produce or commission extra software. The whole package is then marketed as if it were the manufacturer's own product.

A value added reseller is rather different, being typically a dealer who buys in all the hardware often by arrangement or exclusive agreement with a single manufacturer, and then sells it on, with perhaps proprietary software and training included. The value added reseller will specialize in a particular application area, such as accounting, architecture etc.

A system house is a more autonomous body because it does not carry out purchase and resale activities. It is exlusively a service sector body. It can advise on purchase, design systems, and carry out the necessary programming. This software production is thus typically done to order, meeting the needs of a particular client. It may, if successful, also lead to the production of a package sold again to other customers. The transition from system houses to software companies is thus by no means unusual.

The computer user is the remaining category of software producer, which is by far the largest. Computer users, especially of mainframe computers, often develop large quantities of software for use within their organization. Little of it, however, ever moves outside the organization in which it is developed. There are exceptions, however, and a company, for instance, in a quite different field may develop software that is marketable. In so far as its release does not reduce the competitive advantage of the firm, it can be a profitable subsidiary activity. For instance, the McDonnell Douglas Company, which makes aeroplanes, has developed a number of software products which were originally for its own use in engineering design. It now sells these and has moved into many other computer fields also, being a system house as well as software producer. Many other examples can be found of the transition of the computer department of a company, with the latter becoming an autonomous system house and software producer. The phenomenon is more common in the services sector, however, with banks, insurance companies, and accounting firms being typical breeding grounds for such activity.

3. Size and structure of the software industry

Assessment of the size of the software market is very difficult, as is the estimation of the value of software production. The latter question is particularly complicated due to the fact that, as noted above, most software is not traded, but developed within companies or institutions for their own use.

Even considering only commercially handled software, however, the statistical classifications used in industrial statistics are usually inadequate. A particularly intractable problem is caused by the fact that much traded software is produced to order by system houses that will charge the customer not only for the software itself but for a range of consultancy services associated with it, including system analysis and general business services. If the system house is a computer bureau also, it might provide, as part of a package, the actual data preparation and processing. To separate the specific software costs is not easy. A similar problem exists with what is called "bundled" software, which is software sold with a computer as a package deal, or provided free to purchasers of a particular computer. International trade in software is equally difficult to measure because of the similar lack of adequate classifications as well as the different treatments of software for customs valuation purposes. The ease with which software can be transmitted by telephone lines, for instance, further complicates the measurement of its international trade. Finally, the problem of software piracy makes it difficult to envisage any fully statistically consistent pictures of production and trade in software. The spread of pirated software is due principally to the increased demand for packaged software for the millions of personal computer users.

In spite of these caveats, however, estimates of the value of software can certainly be found. OECD has been particularly active in this field, and published a first survey in 1985. It drew on a wide variety of data sources but was unable adequately to separate out specific software activity from the services provided by system houses. A new study [6] gives some alternative estimates. Broadly speaking, the more country coverage is increased, the less precisely is it possible to confine the figures to software alone. Thus, for a total of 32 countries, the software market in 1984 was 26.6 billion United States dollars, and in 1987 it was estimated to total $48.8 billion for 29 of those countries. The country coverage could be wider, but then it would not be possible to separate out all software from services or hardware sales.

United States estimates give the world software market as having a value of $30 billion, of which United States suppliers have an approximately 70 per cent share. Packaged software revenues amounted to 63 per cent of the United States total revenues in 1985, and packaged software for personal computers was the fastest-growing segment. Other estimates include those of United States revenues from overseas sales of software, being over 20 per cent of the total revenues, thus amounting to about $4 billion. Packaged software is 30 per cent of the United States total exports. The largest software markets for United States exports are given as Canada, Western Europe and Australia [7].

Another world estimate comes from an authoritative private source, and gives "software costs" as "$140 billion world-wide" in 1985. The figure was given in the context of a discussion of software productivity, and clearly includes in-house development of software. Reference was made to a present growth rate of 12 per cent per annum, and the world total was projected to be $450 billion in 1995, with the United States share remaining at 50 per cent [8].

Perhaps because of the statistical difficulties in coverage of the total software market, it is easier to obtain estimates of packaged software alone. Thus, one estimate gives a world market projection of $22.3 billion in 1989, having been $5.8 in 1984. Of this packaged software, the share of applications software is 72 per cent in both years [9].

The structure of the industry is a complex one. Certainly there are some large firms, and the tendency in recent years, especially in the United States, has been for them to expand further by acquisition. In the first six months of 1987, there were 137 acquisitions or mergers in the computer services field (here including companies in the software business). The associated value of these agreements amounted to $2.1 billion. In the previous full year, by contrast, there had been only 130 such arrangements, with an aggregate value of $1.9 billion [10].

There are reasons for this tendency, the principal ones being the human capital assets of software companies. A team of unique talents will be attractive for a take-over because they will bring benefits to the acquiring company in a way in which a simple increase in recruitment would not. In the specific case of packaged software, a take-over will allow the acquisition of a product for which the development of something comparable would take years, and for which success might be doubtful. But the single factor most influencing the tendency towards mergers in the software business is that those within it are best placed to identify opportunities. A software company is more attractive to another one than to a financial conglomerate.

4. The role of standards

Two types of standards are found in the software industry. The first is the formal standard established by a national or international body. The second is the *de facto* standard, which is a result of a particular piece of hardware or software becoming so widely used that it is recognized as having created a market. Examples of formal standards include those adopted by such bodies as the American National Standards Institute (ANSI) and those of ISO. In many cases, and particularly with ANSI standards, a national standard goes on to become internationally accepted, with perhaps a version of it subsequently being adopted formally by

ISO.* Examples of de facto standards include MS-DOS and the related PC-DOS, both developed by Microsoft Corporation for the IBM personal computer and compatibles. These are operating systems, and therefore they effectively determine the environment in which software to be used on the computers has to be written. While being extremely widespread and having sold in the millions, these standards have not been adopted by any body as such. Yet they have influenced the creation of thousands of other software products intended to be sold to those who use this operating system on personal computers.

A software standard is a definition of a concept, to be followed by those who implement it as software. Thus, the standard for a programming language is a full definition of the language, its structure, its grammar and syntax, its statements and commands. A software producer can produce a compiler which exactly reflects the idea of the language, and can then describe it as "a full implementation of the XYZ standard". Another software manufacturer can produce a piece of software written in that language, using only that precise version of the language defined in the standard. In principle, therefore, the customer can buy the language compiler from one company and the application programme from another, if they both follow the same standard. If they do not, however, and therefore, if the customer has no guarantee that they will work together, he may decide to buy neither. Languages and application software are a simple example. In practice, most application software is not distributed with the so-called source code, because this is often regarded as proprietary information of the vendor, and to allow others to have access to it would allow them to modify it rather than coming back to the original vendor for help, and thus further business. It would also allow other companies to imitate some of the particular tricks and embodied skills within the software. More practical examples will be found in the area of compatibility between application software and operating systems, or between two application programmes. With respect to data compatibility, for instance, the highly successful package LOTUS 1-2-3 stores its data in computer files in a certain format. Another software manufacturer may advertise his product as able to read files in LOTUS format, thus, this product will be immediately attractive to those who already have the LOTUS product. There is a curious secondary effect, in that the very fact of a manufacturer proclaiming a product to be compatible with LOTUS generates a further support for LOTUS as a standard.**

In general, the issue of standards is very important for software producers, particularly those beginning in the business.

---

*See also [11]. For an illustration of the official standards development process, see [12], p. 96.

**See [13]. Many successful commercial software developments can also lead to official standards subsequently, such as IBM's data base access language SQL, which is the subject of a proposed ANSI standard.

To have a clearly defined standard in the area of application is very important because it provides the producer with some sense of the market for the product, and it provides a relatively stable technical environment in which software development can take place. The producer can be reasonably assured that if the product is a failure, it will not be because of its technical unsuitability, but for some other reason. In principle, therefore, standards can be just as useful to the software producer as they are to the software consumer. However, it is a fact that the setting of formal standards is a complicated process of national and international negotiation, involving many scientific committees at different levels. Typically, the major producers are represented on at least some of the committees involved. They are in a position to influence to some extent the adoption of the standard in such a way that it includes areas in which they have particular competitive advantage or particular skills. In any case, by being involved in the decision-making process of the setting of the standard, they have access to information as to the form the standard is likely to take. This gives them a competitive advantage which is not enjoyed by the small and new producer of software. From this point of view, therefore, software standards are a disadvantage and have severe draw-backs as far as the new producer is concerned. However, it is clear that the benefits of standards outweigh the disadvantages. The question must therefore rather be how best to mitigate the disadvantages from which the small producer suffers. As has been seen, the setting of de facto standards is something with which the small producer is not involved at all, and the exploitation of the market opportunities they create is even more difficult than with formal standards, since detailed information on the product which is creating the standard may be very difficult to come by.

C. Effects of hardware trends

1. Processors and memories

The processor is that part of a computer which carries out the programme instructions that constitute software. Thus it will typically carry out arithmetic tasks such as addition and multiplication, together with other actions such as comparison, branching and looping. These latter and similar tasks are particularly important because they allow the machine the flexibility in hardware terms that is consonant with the generality of software.

The kinds of tasks that the processor can do vary widely in number. Two seemingly contradictory trends exist in this field. One is to make more and more sophisticated processors capable of a large number of tasks. This is supposed to make software development easier, since the machine language programmer then has access to a number of powerful tools. The other trend is to reduce the number of instructions understood by the processor. Broadly speaking, the fewer the number of instructions intelligible to the processor, the faster that processor can work, or perhaps more correctly, the faster that software written for that processor can work. Such processors with reduced instruction sets are called RISC processors.*

The microprocessor is a processor on a single chip. The design trends noted above are intended to contribute to the speed of computing. Another way to do this is to speed up the processing itself, and a variety of technological approaches are used for this. Increased integration of itself reduces the delay in instructions being acted upon. The selection of new materials such as gallium arsenide can also increase the speed, but this means new approaches to manufacturing, and gallium arsenide technology, because of its cost, has up to now been mainly confined to military applications. Increasing the amount of information that the processor can handle at any one time, its basic working unit, is another way of increasing the processing speed. Thus, the earliest micropressors were 4-bit (such as the Intel 4004), and these have been succeeded by 8-bit (such as the Zilog Z80), 16-bit (such as the Motorola 68000) and 32-bit (such as the Intel 80386).** Thus, in terms of internal architecture, microprocessors have reached the same level as minicomputers, for which a 32-bit processor is usual. Another aspect of this is that it makes it easier and thus quicker to handle large amounts of computer memory.

Not only has the facility with which large amounts of memory can be handled increased, but the speed of the memory (that is, the speed with which information can be stored or retrieved) has also grown. Allied to this has been a fall in the unit price of main memory, a fall which has been steady for many years. A temporary shortage has caused a departure from this trend, but the underlying tendency continues to be downward, an inescapable consequence of technology development and the search for a differentiated product, with both these taking place in a fiercely competitive environment.

The consequences for the software market have been very striking. In fact, software can be said to have been struggling to keep up with what has been happening in the hardware field. The capabilities of new computer systems, given the processor and memory trends described, are of a different order to most of the software available for them. This is partly because of the general need to have access to the hardware in order to develop software that fully exploits it. The software development can take several years.

However, a trend in software is towards more "user-friendliness". This means, in general, that computers become easier

---

*For detailed descriptions of some RISC processors, see [14]. However, the article referred to does not cover important new RISC processors such as the Sun SPARC and the Motorola 8800.

**In fact, microprocessors do not always fit neatly into one of these categories. The Intel 8088 was the heart of the original IBM personal computer: it had a 16-bit calculating capability but an 8-bit data bus.

to use. More of the user's mistakes are corrected, more "help messages" are given, there is a tendency for the user to be presented with a range of options rather than be compelled to remember a series of cryptic commands. Growing computing power makes it possible to bring all these ideas to reality. A package has to be elegant, slick and well-planned; the "surface" of the software has to be smooth. No matter how good the central idea of the package is, it must still be well-presented and easy to use if it is to have a chance of commercial success.

The availability of faster processors and more memory means that the software in turn must take full account of it; therefore, as well as "user-friendliness", a second consequence of hardware trends is for software to have more functions or features. The integrated package is one which combines what was previously regarded as separate tasks for software. An integrated package will allow for word-processing, graphic display, spreadsheets, data base etc., all of which used to be separate packages. Even where they remain separate, the competitive package will have extra features: a word-processor will have facilities such as a spelling checker and outlining, for which earlier separate programmes would have had to be bought. The reason for the growth in integration or what is called the "increased functionality of software" is the availability of space and speed in the computers that use it.

Recently a class of materials has been discovered which exhibit the property of superconductivity, that is, of offering little or no resistance to the movement of electrons at temperatures well above absolute zero.* This will have important effects on the future micro-electronic components, and it will mean a greatly increased speed of operation for processors and memories, as well as bringing great changes to telecommunications and most other informatics fields.

2. Architecture

We have spoken as though there were a single processor in a computer, and that is the traditional picture. The so-called "von Neumann architecture" is the internal design pattern roughly followed since the inception of modern computers, where a single processor performs all the calculations and makes all the decisions, one step at a time. In fact, however, this traditional view has been gradually modified. Mainframes and minicomputers commonly have more than one processor, and even microcomputers, although they have one microprocessor, will often have other processors to control screen graphics or floating point calculations. However, it remains true that these processors do not have equal status, and that one processor provides the main control.

Recently, hardware development has concentrated on the linking together of several processors, perhaps very many to work in

---

*See [15]. Superconductors will in the longer term have significant effects on the energy field also. See [16], pp. 19-20.

parallel. There are conceptual and practical problems, but many of these can in principle be solved by software. New languages and operating systems can allow for a new kind of computer programming which takes advantage of the parallel processing opportunities the new hardware configuration provides [17].*

A further development is in so-called neural computing, which is an attempt to follow what is believed to be the way the brain itself works. Rather than the binary logic at the heart of present-day computers, the target is the multitude of interconnections such as are found in the brain, an. the switching processes based upon the attainment of a sufficient number of control signals at each step. The reason for the interest is ultimately the hope of attaining in hardware terms some of the useful characteristics of human thought processes. While hardware development has still a long way to go, some of the features can be explored in software terms.** In particular, ideas of multiple association, such as the human brain is so readily capable of, have already influenced data base design, as can be seen in the development of hypertext systems.

Hypertext systems are a means by which information can be stored with arbitrarily complex links between its components. Thus, a paragraph of text, for instance, instead of being stored under one keyword or even 10 keywords, could have everyone of its words as a keyword and the relationship between the information contained therein and any other piece of information could be followed there through a chain of links. It is this multiple connectivity of information which is analoguous to the way which the human brain stores information.***

3. Peripherals

Several striking developments are taking place in the field of peripherals. With respect to mass storage, a notable tendency has been that of the growth of optical storage systems. These are so called from their use of a laser to read and also perhaps to write to a storage medium. The advantages over the conventional magnetic storage systems such as hard disks, floppy disks and tapes are that they provide a much denser way of storing information, that they are not subject to loss of data from magnetic fields etc. A single CD-ROM, which is the size of a standard audio compact disk, can hold 600 megabytes of information, roughly the equivalent of 1,500 floppy disks of the standard size. A CD-ROM can be written only in the factory. This makes them best suited for information systems which are relatively permanent in nature, typically including such

---

*See also [18] for some commercial issues in parallel super-computing and the role of IBM and Cray. On some differing views on the future of parallel processing see [19].

**For a simple summary, see [20].

***For an overview, see [21]. See also [22].

applications as legal information, statistics, encyclopedias etc. A more flexible type is the so-called WORM which is an abbreviation for "write once read many times". This is also an optical storage medium which can be written to once by the computer user. When a WORM disk becomes full, it is simply set aside.* The full availability of optical disk storage awaits the arrival of a disk which can be written to and erased by the computer to which it is attached. Such a technology is still being developed, but is expected to be brought to market in the near future. The potential of these storage systems is immense. They offer a means by which enormous amounts of data can be stored and easily accessed by a computer user. This means the disappearance, in years to come, of slow techniques such as storage on magnetic tapes of less frequently used information, a practice which is called archiving. They mean also that it will be perfectly practicable to store in character form every document generated by the business, and thus every piece of information in the business could be searched for and analysed by computer, including, for instance, all correspondence over a period of as many years as it is desired. The introduction of scanners, which can read the type characters from paper into the computer, means that it will be possible to store all correspondence in this way, eventually including handwritten correspondence.

Such technology will provide enormous opportunities for software developers. The opportunities will lie principally in providing tools for those who use this information, in order to allow them to make sensible use of it. It is one thing to be able to access quickly all of the relevant information, it is another thing to be able to decide what is important or what is not. The user will still have to search through the material available. Hence "intelligent" software which will help the user to make best use of all the information is a promising field. It represents a qualitative change from the kind of software which manipulates the information, as present-day data base systems do.

The so-called "user interface" has seen developments in several areas. Particularly important is the growing use of graphics. It can be expected that the screens of computer work stations will become larger, will have higher and higher densities, and will provide more and more information to the user. The growth of such techniques as "windowing" allows for the display of several different information areas on the screen, including information about perhaps different processes under way. The increased hardware speeds mean that the numerical calculations involved in graphical displays can be carried out in more and more detail, and this has led to increased sophistication in CAD/CAM systems, for instance,

---

*A 14-inch WORM disk holds 6.8 gigabytes (6,800 megabytes) and typically costs $750 (1987). The 5.25 inch (13.33 cm) size, holding 0.8 gigabytes, is becoming popular in "jukebox" systems. It costs $125. CD-ROM unit costs are much lower, but there is a fixed cost of about $15,000. See [23]. However, mastering costs as low as $1,500 and duplication costs of $2 have now been reported [24].

where diplays which can manipulate three-dimensional images are more and more usual. The software developments associated with this include the growth of what are called object-oriented languages, since the traditional mathematical languages such as FORTRAN, often used to produce graphics images, are inadequate for the kinds of sophisticated graphics application now being developed. A further push towards object-oriented languages comes from the growth in artifical intelligence applications. Thus, hardware developments in display technology have created a significant market for software products, and one that will grow.

Communications between work stations and computers and between computers themselves is another growth area where hardware developments have allowed for the creation of very elaborate networks. These networks in many cases span countries and continents. Where telecommunications have been liberalized, and this is an increasing trend, the linking of computers has become commonplace. Fibre optics developments mean faster communications of this kind. There are, however, still very significant problems in data communication which the emergence of new standards has not yet overcome. Many different protocols exist, different computer manufacturers will follow different standards, and national telecommunications authorities in many cases restrict the use of standards to those specified by them, or refuse to allow the connection of any equipment to the national network which has not been certified by them. This means that communication software in general has many compatibility problems to overcome. There remains also considerable scope for improved networking standards which would allow for more flexible and faster communication between different systems. An enormous number of niche markets is thus created, since software to overcome particular compatibility problems, or to allow two pieces of equipment to be connected together, may provide a neat solution to what is an otherwise intractable problem or, alternatively, a relatively simple problem which has been ignored by the makers of the equipment. Thus, even the confusion in the computer communications field can bring benefits to the software producer, although it is such a rapidly changing field that these have to be continuously solved in new ways. In the longer term, the trend is towards an Integrated Services Digital Network (ISDN) system, which will be a public network to replace the existing separate telephone and dedicated data networks. Such a tendency will lead to increased standardization and perhaps fewer opportunities of this kind. However, the process is not likely to be completed in a short time. The spread of the Open System Interconnect (OSI) standard is a further development which will influence communications software significantly in the longer term [12]. Again, the transition periods may offer opportunities for the bridging of gaps.

D. Trends in software

1. Overview

In this section trends in software which are expected to have an impact on world markets in both the shortand the medium-term are examined. Software in general can see a long delay between a theoretical advance and a commercial application. This is for several reasons. First, an advance may simply be of no commercial interest. Secondly, the software advance may be rendered redundant before it can be marketed. For instance, a software advance that speeds up a particular calculation or enables less storage to be used may well be irrelevant in a context of rapidly falling hardware prices. Thirdly, software has to encounter significant resistance in its traditional consumers. Computer staff will have invested several man-years in the construction of a particular application. The new product, even if better and cheaper, may still require conversion of the existing data, and time to be spent in a transition period when both systems have to be kept working, the old and the new. It can appear easier to keep the old system in operation. The situation is worse when the product offers a more dramatic break with the past: it may imply data structures and system procedures which are so different from what has gone before that they intimidate the potential user who has been conditioned by the existing system. The producers therefore cannot move too much ahead of the users.

This is not to diminish the role of innovation in the industry, but it must be realized that bottle-necks occur, and the absorptive capacity of the market can be limited. Good ideas may have to wait their turn, especially if their cost is high either in terms of purchase price or in terms of further investment of own resources by the user.

There is nevertheless scope for innovation in software, especially when it is associated with hardware innovation. The best example is given by the microcomputer. Its widespread availability generated a number of software innovations, with products being produced which were qualitatively different from mainframe and minicomputer software. Spreadsheets are a product of the microcomputer era, which might better be called the personal computer era. This is not only because of the dominance of the IBM personal computer and its clones, but also because the personal character of the computer determined much of the software which was written for it. Spreadsheets are targeted towards the manager, accountant or clerical worker who works with tables of figures. At the simplest level, a spreadsheet will maintain the row and column totals of the table, and can be set to automatically adjust the totals when any individual figure or group of figures is changed. In practice, spreadsheets have grown more and more sophisticated: the possibility to specify arbitrarily complex relationships between table entries and between groups of tables, as well as the order and manner in which recalculation can be carried out has meant that spreadsheet commands have evolved into programming languages. Nevertheless, the natural matrix orientation of these spreadsheet languages gives them characteristics unlike any of the widespread high-level computer languages such as FORTRAN, COBOL or PL/1.

A further genre of personal computer software has been the integrated package, where a number of different functions (spreadsheets, data base, graphs etc.) were combined in one and allowed the user, for instance, to move data from the data base to the spreadsheet, to make calculations from it and then to draw graphs to display it. Previously these steps would have had to have been carried out through the separate selection, loading and running of these different software packages. The integrated software package allowed the user to computerize all his traditional office activities. A later development was the addition of other facilities, especially word-processing, within the package.

A further feature of microcomputer software has been its "user-friendly" character. This arises from two causes. First, some of the individuals who pioneered the development of microcomputer hardware and software were partly motivated by an individualistic feeling about the role of computing power in general. They did not see it as something centralized and something that should be accessible only to the initiated. A second and now dominating feature is that the commercial packaged software producers realize that they are catering to a market the majority of whose members have either no computer experience at all or else are essentially self-taught, being accustomed only to using other packaged microcomputer software. The consequence is that they must at least maintain, if not improve on, a tradition of user-friendliness. This trend is reinforced by the hardware trends already discussed in section C above, where the falling costs of memory and storage devices means that software can be bigger and store more messages to be sent to the user, guiding him or her through the use of the programme or package in question and explaining errors, not just pointing them out when they occur. This trend further exemplifies the need for software developments in some cases to wait on hardware developments, even though the inventive character of user-friendly features may in fact be very low.

It is useful to make a distinction, in examining trends in software, between evolutionary change and new technology software. Evolutionary change in software can be defined as change determined by improvements to previous stages of the software, where the lines of descent are fairly visible. In simple terms, improvements have been made to existing ways of doing things. New technology software on the other hand refers to software developments which are associated with using computers to do things previously done in other ways or not done at all. Here can be included most artificial intelligence (AI) applications, such as image recognition, speech processing, computerized translation and expert systems. It can be stretched to include concepts of CAD/CAM and CIM, which amount to new ways of doing tasks which had not previously been automated. The point to be emphasized is that the first computerization of accounting, payroll, stock-keeping etc. took place many years ago. New software development in such fields as these is evolutionary because it accommodates to some degree at least the systems which are already in existence and allows the exploitation of innovation without the perhaps painful abandonment of existing systems.

The line between evolutionary and other software can be hard to draw in some cases, especially when the software product incorporates features which are partly evolutionary and partly not, such as the use of expert systems in data base applications. Why, then, make such a distinction? It is important strategically because different markets are in question and it is important also in investment terms. A new departure in software may, precisely because it starts from scratch, involve considerable inputs to reach production status. Financing the development costs may be painful, since cash flow in the early marketing stages may be slow. The evolutionary product, on the other hand, may represent a repackaging of an existing product or its transfer to another computer system. Software tools such as cross-assemblers may make this process easier. Again, the new evolutionary product may be able to take advantage of elements of the software environment in which it applies, and thus make use of improvements carried out by someone else. For instance, the systems software of a computer often undergoes continuous improvement by the manufacturer. It may contain new and efficient ways of reading, copying, comparing and checking data and displaying it, storing it, and printing it. This sort of software is unglamorous, and yet it involves a great deal of detailed work. The astute independent software developer can take advantage of much of this work and thus spare a good deal of the development time by finding out how these parts of the system software work and allowing his own software package to use them. As well as saving time in the development phase, this will very often mean an increase in the speed of operation of the finished programme itself. Again, some parts of the system software may be inefficient and the producer will decide to bypass them, approaching the hardware directly. Such practices, however, have their dangers for the software developer. The reason is that they can tie the product too closely to the hardware of one computer manufacturer, or to one version of the system software, and thus restrict the potential market and increase dependency.

2. Operating systems

As briefly mentioned in section A, the operating system provides the necessary control and communication between the different parts of the computer and its peripherals in such a way that it is accessible to the user. The operating system provides a framework, more precisely a defined software environment, in which applications software can be written. It also carries out a lot of the tasks needed by the application, for example, control of the input and output, and, in multi-user systems, time-sharing and automatic back-up and recovery. This also means that an operating system defines a software market.

The point is worth emphasizing: the same operating system, if it is available on different computers can, in principle, allow an application written for one computer to run on another. Equally, the same computer can use more than one operating system, and an application written for one computer will not run on an identical computer if the operating system being used is different. The operating system thus insulates the software developer (to a large extent) from the physical computer. But it makes software development less hardware-dependent only at the cost of making it operating-system-dependent. The role of high-level languages in countering this tendency is examined below.

In general, a computer is supplied with operating system software included, since a computer without an operating system is of interest to only a specialized few. Therefore, it has almost always been the case that the computer manufacturer supplies an operating system, usually written to take advantage of whatever hardware features characterize the new computer, but increasingly with an emphasis on distancing the user from hardware considerations per se. At the same time, attempts are made to accommodate earlier versions of the computer and operating system, either by providing that any software written to run under the old system will run under the new one even if it does not use all the new features provided in the latest version ("upward compatibility"), or else by supplying a set of software to assist in the conversion process, that allows the old software to be automatically changed so that it can function under the new system. This is sometimes called an "upgrade path".

Operating systems are sometimes classified as multi-tasking, multi-processing, multi-user etc. In market terms, the most vital distinction is perhaps between multi-user and single-user systems. Broadly speaking, mainframe and minicomputers have multi-user operating systems, either in the sense that the system can deal with a number of different tasks submitted to it, or in the sense that many users are actually physically linked through terminals to the computer.

Multi-user systems are what is needed in ordinary economic activities. Most of these involve exchange of information between those working in the same organization. If the computer contains the organization's data in a multi-user system then all the staff can in principle have access to the data and change it as necessary. It is this relatively simple idea which lies at the heart of the data base concept and explains the growth of this in recent years.* In practice, for most commercial and industrial tasks, multi-user systems are necessary, or soon become so.

It has often been remarked that personal computers, and especially those based on the newer 32-bit processors, are more powerful than many minicomputers. In some senses, that of calculating speed and memory management speed this is certainly true. But in another sense, that of multiple use, this is not so at all. The typical personal computer is not designed to be shared. Only one person at a time can use it,** and it is not even designed to be used sequentially, since there is little or no security to protect unauthorized use or accidental destruction of the software or data stored in the computer. The kinds of software

---

*The movement from a "Ptolemaic" to a "Copernican" concept of data processing is described in [25]. For a general introduction to data base ideas, see [26].

**Multi-user operating systems are available for many models of personal or microcomputers such as XENIX (a form of UNIX) from Microsoft Corporation. However, the overwhelming majority of such computers have single-user operating systems.

available for personal computers reflect this type of individualistic use and in general encourage it.

The gap between single-user and multi-user systems is bridged by networks. A network is a combination of hardware and software, which allows computers to be linked together, allowing for the easy transfer of data between them. Personal computers, minicomputers and mainframes can all be combined in networks in different combinations, allowing for the sharing of information between different parts of an organization using different computer systems. Networks thus provide a way out of the dichotomy between single-user and multi-user systems. Each part of the network can, if necessary, be established as an independent system but with the capacity to exchange data with the other parts.

In practical terms, a form of networking tendency can be seen in the growing emphasis on the linkage of personal computers with mainframes and minicomputers. Work stations of considerable power are replacing the traditional dumb terminals. The personal computer system OS/2 Extended Edition, which is being developed for the IBM PS/2 series of microcomputers based on Intel 80386, will contain specific provision for accessing mainframe data bases and downloading selections from them for analysis on the PS/2.

For mainframe and minicomputer operating systems, a significant trend has been the growth in importance of the UNIX operating system. UNIX is notable because it represents a standard that operates over mainframes and minicomputers from different manufacturers. It is also increasingly available for microcomputers as these expand in power and speed. Thus it offers a bridge between all three categories of computer and in principle allows software developed for one type to be usable on all.

What is UNIX? An operating system is difficult to describe in a few words, but it has one important characteristic, its multi-user orientation to which particular attention was paid in its design. Another important feature is its "piping" facility, which conceptualizes data-processing, input and output in such a way that the user is not concerned with the physical characteristics of devices and can freely and easily change his or her use of them. UNIX is a large operating system because it contains many features that are optional extras on other systems. It thus spreads standards into areas such as security, data transfer and networks. The wide-ranging and comprehensive nature of UNIX thus adds to its attractions.

Its critics say, among other things, that it is a cumbersome system which requires a good deal of main memory in the computer to use it, that its syntax is clumsy, that not enough good commercial applications software is available to be run under UNIX, that too many different versions of UNIX exist, and that the progress in spreading UNIX has been so slow that better operating systems are now available. None of these points are accepted by UNIX supporters, who point out that memory prices have fallen so much that the size of UNIX is no longer an argument against it, that more user-friendly interfaces to UNIX are available, and that whereas in the past use of UNIX was concentrated in academic and the scientific community, now much commercial software is available for business applications.

UNIX is not only a technical phenomenon, however, it is an illustration of the kinds of competitive tensions which characterize the world of hardware and software manufacturers. As such, its history is confused, and its future only slightly less so. At one stage it was regarded as a vehicle for challenge by AT&T (its original developers) to the position of IBM [27]. It has also been seen as a means by which many other (mostly European) computer manufacturers could mount a similar challenge, setting up a corporation (X/Open) to do so.* Another approach through traditional standards-setting seemed to bring consensus on a universal UNIX interface (POSIX),** and involved several competing companies. However, a growing commitment on the part of IBM to its own form of UNIX, called AIX, and a developing relationship between AT&T and Sun Microsystems [31], may have been among the factors leading to the formation of another body, the Corporation for Open Systems, which united IBM with many other companies in a bid to promote open systems based on the X/Open and Posix standards.***

The organization X/OPEN is worth mentioning in connection with UNIX because it shows a remarkable tendency on the part of several hardware manufacturers, particularly European, to combine in what is essentially a market-sharing strategy. The objective is to allow for the free movement of applications software from one make of computer to another.**** By this means a large common market in software can be created. If successful it would greatly assist the survival of the smaller manufacturers of computers. At present these have to go to considerable efforts to ensure that there is a sufficient choice of software available for their computers. Given the increasing development costs of software and the competition

---

*See [28], as reported in Data Processing Digest, October 1986. See also [29]. The first step has been the preparation of the X/Open Portability Guide, describing the common applications environment (CAE).

**Posix is a standard for interfacing with UNIX, set by the Institute of Electrical and Electronic Engineers (IEEE 100 3.1). See [30].

***The Software Foundation is supported by Apollo Computer, Group Bull, DEC, Hewlett Packard, IBM, Nixdorf and Siemens. See [32], p. 62. Its work will be based on the IBM version of UNIX, called AIX.

****The standardization work of X/Open is concerned not only with UNIX but with applications languages, so that an application may be written in a certain form of COBOL or FORTRAN and then run on any X/Open standard computer without the source code having to be changed. X/Open is including SQL (see following section 4) in the next version of CAE.

emerging from the microcomputing world, this is an ever more difficult task. For some manufacturers, a commitment to X/OPEN, and to the development of UNIX as a standard, may well have been the only course to follow.

It should be added that UNIX is not the only operating system available for different brands and types of computers. Another one, PICK, is an operating system with embedded applications, data base management in particular. Traditional distinctions between operating system commands and data base inquiry languages become blurred in such a context. PICK has done quite well in specific commercial fields, but its role in more general applications is as yet unclear.

What does all this mean for the potential software producer? There are two important lessons to be learned. The first is the increasing complexity of operating systems. UNIX exemplifies all this very well. In itself it is complex, with many facilities and many ways of doing things. But as well as this, there has been over the years an accretion of software which adds to or expands on the basic facilities of the system. For instance, the IBM version, AIX, has added half a million lines of code to the original version of UNIX, according to an IBM advertisement in [33].

This trend in complexity is also seen in OS/2, the operating system to replace MS-DOS for personal computers using the Intel 80286 or 80386 microprocessor. MS-DOS was supplied on one diskette, while the Extended Edition of OS/2 will need about 16, but it will include considerable data base and other capabilities [34].

Clearly this could make things difficult for the new software house. To master fully the complexities of new operating systems may be beyond the capabilities of a small company. It may not be able to discover all the tricks and hidden short-cuts in the operating system which can be used to give the package a unique character. Again, one of the reasons operating systems are growing more complex is that, as noted, they are incorporating more and more functions previously regarded as extras to be provided as additional systems software or as applications software. Data bases and data-base access software is a particularly striking example of this. The opportunities for third-party software firms are therefore reduced accordingly.

Apart from complexity, however, another clear trend is towards less diversity in operating systems. The move towards a reduced number of operating systems in the market is very clear. The UNIX trend is only part of this picture. In large-scale mainframes IBM has from 75 to 80 per cent of the total market, and has established the MVS and VM operating systems as the de facto standard. In 1986, there were over 20,000 copies installed (which were estimated to generate recurring revenues of $1.5 billion) [35].

In minicomputers de facto standards are less clear, with many manufacturers having their own proprietary operating systems. Some forms of UNIX can also be found, offered as an alternative to the manufacturers' own operating system. However, IBM's System 3X computers have been predominant. IBM was reported as giving the System-3X installed base as 220,000 world-wide in 1986. The number of application packages available was given as 4,000 [36].

In microcomputers the dominance of PC/DOS and the similar MS/DOS as operating sytems is overwhelming. The only significant alternative is offered by Apple Inc. The new operating system OS/2, already referred to, will soon make an important impact, even though its final form will probably take some time to appear.

The future horizon, therefore, offers some measure of stability for the new software developer. The trend towards fewer operating systems means larger potential markets. The hardware trends which may ultimately lead to the disappearance of the traditional minicomputer will further reinforce this.

3. High-level languages

High-level languages are so called because they allow the user to communicate with the computer in a form closer to that used in normal communication. High-level languages are increasingly important in the software world because of the hardware advances referred to, and because of the increasingly competitive character of the software market. Each of these points will be dealt with in turn. Since hardware change has meant faster processors and memories, it means that it is more practicable to consider using a high-level language in developing a piece of software. Previously, to get sufficient speed in the final product it was often necessary to use machine code or assembly language. This is in spite of the fact that most programmers prefer to work in high-level languages, and it is quicker to develop and test programmes in such languages. High-level languages also have special characteristics which make it less likely that the programme will contain mistakes.

Increased competition in the software market has given a further impetus to this trend, because use of a high-level language can shorten the development time and can give access to a wider market. While a computer-maker probably offers a slightly different version of a high-level language, the conversion from one system to the other can be straightforward enough, provided the software developer adheres to some minimal definition of the language and avoids taking advantage of too many of the special features offered in the computer manufacturer's compiler.

In general, the high-level languages in use are very old. FORTRAN and COBOL have existed in different versions since the 1960s. They maintain a strong user base and formal standardization procedures continue to add new features and improvements to them. The rise of the microcomputer saw a revival of BASIC, first developed also in the 1960s for educational purposes. Business use of BASIC appears to have received an impetus from this. However, there has been a lack of standards and a divergent growth of versions of BASIC far more sophisticated than the original. The spread of new languages is also significant, however. These include C, Ada, and Pascal.

C is the language in which the UNIX operating system was written. It is thus scarcely new anymore, and yet in recent years has grown rapidly in popularity and has moved outside the UNIX community. Its virtues are several. It can attain in many cases the speed of an assembly language while having most of the features of a high-level language. It allows operations which are characteristic of assembly language and machine code, permitting access by the programmer to many aspects of the operating system and hardware devices. This makes it suitable both for systems programming and for process control and other real-time applications. The other advantages of C include a number of sophisticated features such as pointers and data structures (as in Pascal) and list manipulation features (as in Lisp). For all these reasons, the language appears to be being more and more widely used.*

Ada is a language developed for the United States Department of Defense, which is now coming into wide use, including outside the military field. The objectives of Ada were to have a single language that could be used for all embedded applications. It would thus replace the use of assembly languages or specialized languages for all real-time applications such as weapons control. All contracts awarded by the Department would stipulate the use of Ada. Once the standard had been defined, software manufacturers began to prepare Ada compilers, the Department of Defense set up a committee to "validate" compilers, that is, to confirm that they met the required definition of the language, and a large number of compilers for different computers have already been validated. Thus, in contrast to all other high-level languages, Ada's development has been carefully controlled and standardized. The use of Ada is another question: there are indications that because of the very wealth of facilities available to the programmer in Ada, it is perhaps unsuitable or at least unnecessary for many programming tasks. This must mean that, at least in civilian applications, the scope for the use of Ada may be limited. However, interest in Ada and in related development has already extended far outside the United States. In the future, as skills diffuse through the commercial computing community, preferences may emerge for the use of Ada if the programmers' experience of what is undoubtedly a powerful applications language is a positive one. Accordingly, the conclusion must be that Ada will be an important influence in the commercial field, but that this is still some years away [39].

Two other languages, Pascal and Modula-2, are both inventions of the Swiss scientist Nikolaus Wirth. Pascal has established a significant user base. It has advantages in many application areas, in both commercial and scientific fields, although its success has been greater in the former. An important feature of Pascal has been its adoption by many academics as a more suitable language to illustrate good programming techniques. The theoretical trend towards what is called "structured" programming can be traced to the work of Dijkstra ([40], [41]), who has laid down basic principles

*For a review of the C language, see the articles in [37] or see [38].

of the discipline. Pascal's advantage lies in the provision of a large number of control structures which allow for the application of these ideas in practice in a way that other popular languages cannot do. By meeting several of the criteria laid down for structured programming it found ready acceptance among those who sought the reliability or verifiability of computer programmes designed according to these principles. Pascal is therefore a case where academic theory did indeed provide a certain impetus for commercial popularity.

Mainframe acceptance of Pascal has been less. Most major computer manufacturers offer a Pascal compiler for their machines, but its use in commercial data processing appears to be limited. The reasons for this are that when change is contemplated (such as the replacement of an old system written in COBOL) the designers are likely to be more attracted by the increasingly common C language or by the features of a fourth-generation language, since it may already accommodate features (such as screen-handling and a database query language) that are not part of the definition of Pascal or of other high-level languages.

Modula-2 is intended to replace Pascal and to remedy its observed defects. Its commercial prospects look mixed, partly because of the wide use now being made of Pascal in those markets which might have used it. However, its longer-term influence on language definition and the next generation of computer languages may be considerable: the true modularity possible in it, and the power of the user to define communications between different modules, are features which are likely to be seen as very desirable in any future definition of the correct functioning of a computer language.

4. Fourth-generation languages

Fourth-generation languages (4GLs) are proposed by their producers as a replacement for the traditional data processing languages (such as COBOL, RPG, PL/1 etc.) which are regarded as third generation. The second generation would include assembler type languages which provide mnemonic and limited macro facilities, one step removed from the first generation, the machine code used by the first computers.

Fourth-generation languages have to be distinguished from languages such as Ada, which provide a number of new concepts for practical use by programmers. Fourth-generation languages are intended to reduce programming: the statements and commands encapsulate many lines of COBOL or PL/1 code. Operations that might take a large amount of space and time to specify in a third-generation language can be conveyed in one or two lines, which are often in a form that is nearer to English.

What might be called a subset of fourth-generation languages is the programme or application generator. This is essentially a compiler whose output is a COBOL program. Give the set of commands which it is desired to implement, the program generator then produces appropriate COBOL statements. The advantage of this is that the code is, in principle, portable to another COBOL environment. Another advantage is that the code produced can be examined and directly modified in a way that is not possible with a normal compiler whose output is machine code.

A number of features may be part of the fourth-generation language, including screen-handling, data entry, printing formats, data base management, access and updating. All these aspects, if they are included, will be distinctive. Standards, whether formal or de facto, do not exist at all in this area. Experience gained in one fourth-generation language may not be of much use in another. Here is a field, however, where competition is increasingly fierce between different groups. The computer manufacturers now often offer some form of fouth-generation language, but the large software companies are also extensively involved.

The future for fourth-generation languages is promising. The driving force is the shortage of programmers and the limited increases in their productivity, combined with increasing pressures to computerize as many aspects of commercial life as possible. Competition combined with the increasing complexity and changing character of commercial data to force companies to improve and expand their information systems. However, the typical effect of such pressures is a growing backlog in the data-processing department of a company. Fourth-generation languages appear to offer a fairly easy way out of this difficulty, since their main claim is that applications can be developed far more quickly than by traditional methods. The drawback is that they require a certain commitment of resources, a certain amount of retraining of staff, and, most importantly, an abandoning in many cases of patterns of work which have been in use for perhaps 20 years or so. In addition, the very pressures on data-processing managers will tend to limit their scope of action. Concerned with maintaining, modifying or patching up existing systems, they will not have time properly to examine the alternative approach to problems offered by the fourth-generation language.

The other highly significant feature of fourth-generation languages is their convergence with data base management systems (DBMS). The fourth-generation language may offer a database management system of some kind as an integral part of the language or "environment", a somewhat vague term used to describe anything from a user interface to a suite of programmes marketed by the same producer with some commonality of command structure and syntax. Alternatively, the data base management system may offer sufficient facilities in terms of data input and output and report writing, as well as data storage, so that its command structure is in effect a fourth-generation language.

A good illustration of the convergence is the spread of the SQL language. This is used to access data from relational data bases. Developed by IBM, it has now become part of the X/Open standardization effort as well as being defined in an ANSI standard. It is now available for many data-base management systems on mainframes, minicomputers and personal computers also.

5. Data-base management systems

The growth of the data-base management system, and, just as significantly, the broadening of its definition, may be seen as a process where software has developed to meet the practical revealed needs of computer users. The third-generation language FORTRAN grew out of the need to express scientific relationships in familiar, formula terms. COBOL was developed for the United States Department of Defense, but it grew again out of the need to express typical commercial calculations.

Third-generation languages deal with atomistic data. They do not have built-in methods of handling lists, or collections of data of different kinds such as "name, rank and serial number". Equally important, the storage as such is not defined. The languages are intended to process data, that is, to read it into the computer, do something to it, and then write out the result, whether to a screen, to a disk drive or to a printer. Even on-line systems can be just a straightforward extension of this particular concept.

The data-base management system provides a means of storing data in such a way that some, or in some cases all, of the relationships within the data are preserved and can be followed through in searching it. It also provides a way in which the data can be selected according to different criteria and summarized. Typically, this summarization procedure will be in the form of printed tables of results. In this case the stage is called "report generation". The data-base management system may also provide a wide number of other facilities. These facilities can include a number of ways of dealing with the input of data, including its specification with respect to form and content in such a way that only data of the correct type can be entered, and only in certain circumstances and by persons authorized to add or amend data in the data base. The facilities can also include a good deal of "housekeeping" functions invisible to the user or the constructor of the data base, but nevertheless important in ensuring that the storage media (for example, hard disks) are used in the most efficient and economical way.

Data-base management systems can be classified in many ways, and they range from systems essentially analogous to card-files of the conventional kind to other systems of an elegance and complexity sufficient to exceed many ordinary commercial needs. A common classification scheme is to divide data-base management systems into three categories, hierarchical, network and relational. It is the last category which is the focus of most new development and competition.

Relational data bases are so called because they store the data in such a way that connections can always be traced and revealed as required. The origins of relational data-base management systems can be traced directly to the famous article by Codd, who set out criteria for the establishment of the relational principle and pointed out that no existing system met these requirements.*

The basic idea of a relational data base is that it treats all relationships as composed of pairwise relationships. This allows all data to be stored in the form of two-dimensional matrices (that is, tables). Depending on the data, it can therefore appear in more than one table. At first sight this duplication (or indeed multiplication) of storage requirements appears very wasteful. However, the design of the data-base management system itself can minimize any duplication, and in any case with the rapidly falling costs of hardware this criticism will tend to be less important.**

In spite of the difficulties associated with transferring existing systems, relational data-base management systems are now making a considerable impact. The main reason for this is not so much the perceived virtues of relational technology but the pressure for the systems in general. The information within an organization is increasingly seen as its most important asset, and easy and flexible access to it is required. The role of data-processing departments is tending more towards the construction and maintenance of a data base and the control and facilitation of access to it. Thus, data-processing becomes a continuous task rather than the series of discrete "jobs" into which it used to be decomposed.

6. Computer-aided design

Discussion of CAD is often bracketed with that of CAM, although in practice it is usually CAD that is being talked about when CAD/CAM is under review. This is because of the role of the computer in automating the process of design, a fairly clearly delimited sphere. Roughly speaking CAD/CAM means in practice the automation of the design process in manufacturing. When CAD is combined with a wide range of diverse informatics techniques such as robotics, remote sensing, process control, numerically controlled machinery, automated stock-handling, ordering and inventory management, then, provided all these processes are linked in a common communication and control system, the whole approach is known as computer-integrated manufacturing.***

How can design be automated? Strictly speaking it is not the creativity at the heart of the design process that is automated but rather a number of mechanical tasks which the designer has to carry out in order to realize his or her objective: a set of drawings and

---

*These criteria were set out in Codd [42]. More recently, Codd is reported as having increased the number of rules to 166 features grouped in 13 classes. See [43].

**Except in so far as it contributes to a slow-down in access times.

***For definitions of CAD, CAM, CIM etc. see [12], annex I, pp. 149-158.

instructions for tooling up for the construction of the required object. The designer draws on the screen of a computer terminal. When the result is satisfactory it can serve as the direct input to production (for example, by controlling the movements of a machine that cuts or shapes some material according to the design that has been achieved). More simply, the design can be produced as a drawing to be used by the production designer.

Drawing on the screen is accomplished by allowing graphical input from the designer and using a number of software and hardware tools. The graphical input can take several forms: a digitized image of a hand-drawn sketch can be used, or a joystick, or a digitizing tablet, a mouse, or cursor keys. These treat each point in the drawing as a series of co-ordinates to be afterwards manipulated, the rough initial sketch being refined on the screen to a finished design. Facilities for doing this vary with the sophistication of the hardware and software, but include for instance, the automatic joining of points (line-drawing), enlargement along a specified axis, rotation about a specified axis etc. The generation of three-dimensional objects (viewed, of course, in two dimensions on the screen) is also included (with a selection of basic shapes provided as building-blocks for the image). This allows an object to be examined from any desired point of view, and can be a means, for instance, of estimating the actual appearance of an experimental design, and thus replacing the construction of solid models as may have been common practice in many industries. A cup, a table, a box may, as a computerized image, be modified and re-examined and appraised in a way that allows the designer rapidly to assess the acceptability in production or consumer terms of a proposed design. Another example from architecture is a package which allows the user to design a building and "walk" through it: given the structure of the proposed building the computer can calculate how it would look not just from outside but from any position within any room of the building also.

It is not perhaps surprising that some of the most sophisticated applications of CAD, in fact true CAD/CAM applications, are found in the electronics field. The design of electronic circuits on a screen, with such features as automatic minimization of interconnections, is now standard. The output can range in sophistication from a printed design, to data in a standard format, to specific control sequences for the machines which manufacture the chip [44]. Even in such a highly specialized field however, there has been a trend towards standardization: a growing commitment to, for instance, the international graphics exchange standard (IGES), the manufacturing automation protocol (MAP) and OSI standards, together with such de facto standards as the Gerber data format and the Sun work station as a "platform". The growth of technical work stations has meant the increased availability of large sophisticated processing power in individual units. It thus expands the potential market, in unit terms, to a considerable degree. This has provided an important stimulus for the creation of software companies in the design field [45].

7. Impact of artificial intelligence techniques

(a) Overview

The development of AI techniques is already having important effects on the world software industry. While debate continues at a theoretical level as to what is AI and as to whether there are limits to its future development, it has as its main concern making computers replace some primary human function such as sight, speech or analytical thought. An alternative definition is the following:

> "AI is concerned with programming computers to perform tasks that are presently (sic) done better by humans, because they involve such higher mental processes as perceptual learning, memory organization and judgemental reasoning." [46]

AI techniques are influencing the following areas in particular: speech input and output; pattern recognition, and expert systems. Why is AI becoming increasingly important? The first reason is that the enhanced capability of computers makes possible the application of many AI techniques which rely on a good deal of processing, typically in the form of searching for a matching pattern between input and some stored data base. In this sense the software development is driven by developments in hardware. There is also a growing demand for applications in speech and character recognition, where the object is to automate the process of data entry. This trend is encouraged by the growing availability of enormous mass storage in the form of optical disks.

However, a definite if difficult to measure factor influencing the growth of the AI market is the push from software producers. Many of these are casting round for new product ideas. The personal computer market is saturated with spreadsheets, word-processing packages and integrated packages of various kinds. The mainframe market still has many possibilities for increased sales in the areas of data-base management systems and fourth-generation languages in particular, but this market is weighted heavily in favour of the computer manufacturers and of larger companies that can field a sales force and provide service and support to a fairly conservative market. Thus, the search for new product ideas enters the area of AI. This is not, of course, the only area in which activity in the development of new software products is taking place. However, it appears to many as the area which offers particular scope for innovation and promises great rewards for success.

Other factors influencing the growth of AI include advances made on the theoretical side by researchers, especially in universities, who find the subject and the problems interesting. Specialized demands, such as those by international organizations for automated translation and interpretation, or the need to deal in data-processing terms with a representationally complex language such as Chinese, also combine to give a further impetus to research in this subject.*

Robotics and the automation of the manufacturing process are another cause of the growing interest in AI applications. The constant search for an improved competitive position and, especially in developed countries, the need to minimize total labour inputs to the production process have caused accelerating interest and applications in this field. Other factors at work in the spread of automation of the production process include the need to minimize use of materials through more efficient cutting, measuring etc., to speed up delivery times, to meet a wide variety of demands for differentiated products, and to meet rapid changing demands or changes in the relative costs of inputs.

Robotics and automation are not per se a branch of AI, but they increasingly use AI techniques, especially in such areas as sensing. This means that machines are made to recognize objects or characteristics of objects, either through direct contact or by vision, where a digitized television image of the object is captured and tested as to whether it matched a pattern already known to the machine, and action is then taken accordingly.**

Automation also includes the speeding-up of still manual tasks of control by means of speech input and output. For instance, instead of having gauges and indicators, a machine could "speak" the temperature, pressure or whatever. In difficult industrial conditions this may greatly improve the carrying-out of specialized tasks, since the operator is not distracted from them by having to monitor a dial, a graph or a digital read-out. Similarly, speech input can allow control of a machine or a process without the operator having physically to touch a button or a switch. Even simple processes such as stock-taking can benefit from speech input, since the stock-taker can simply call out the numbers and types of the items being counted, rather than entering them at a keyboard, using bar-code readers or any other direct handling of the objects or equipment.

(b) Expert systems

Expert systems are the branch of AI applications in most widespread use. They are systems which attempt to embody human knowledge and expertize in an accessible form, typically leading the user through a series of questions to resolve a problem ([46], [50] and [51]). Expert systems are now a commercial reality. All the main computer manufacturers offer products in this field. The pioneering academic centres of expert system work, such as

*Automated translation has been a particular focus of interest at the Commission of the European Communities, with the Eurotra project (see [47]). For a survey, see [48].

**For a detailed account of pattern recognition techniques and applications, see [49].

Carnegie-Mellon, Stanford, Edinburgh and Marseilles, continue to lead the university world in the subject, but the bulk of activity (often hidden) is in the commercial field.

In hardware, it has been noted that many purchasers are turning away from dedicated AI work stations, and instead increasingly taking the much cheaper option of buying a software package to run on an existing personal computer such as an IBM PC or equivalent [52].

In software the proliferation of expert system shells continues. A shell is the software needed to construct an expert system, and written in a generalized form so that it can in principle be used for any subject area. Shells are offered for sale at prices from several hundred dollars upwards, and new ones almost invariably for microcomputers rather than for minicomputers or mainframes. However, they vary very much in quality, sophistication and ease of use. Users also continue to develop their own expert system software. For this purpose the PROLOG programming language has made considerable progress. However, the United States dominance in the commercial expert systems field means that the LISP language still is the major one, since a heavy investment in this language has been made by software developers and computer and work-station manufacturers.*

The use of expert system shells will continue to increase, but it is likely that a plateau will soon be reached. This is because some disappointments can be expected: the naive business user who purchases an expert system shell to be used on a personal computer will gradually realize that he has bought merely a tool, and that the task of constructing a useful expert system is a fairly long and demanding task that still lies ahead of him.** New shell products can in the immediate future offer improvements only in fields where progress has from some points of view been already sufficient to be continued, for example, in faster inferencing, in more detailed explanation of the processes by which decisions were arrived at, and in the incorporation of more user-friendly techniques. Progress can also be expected in the area of constructing the knowledge base, allowing more flexibility in the specification of rules, for instance. But the real problems of expert system construction remain, strictly speaking, outside the realm of software. The decisions as to how to limit the problem area, how to ask the right questions of the human expert and how to decide what are the important parts of his answers are difficulties which the present levels of software development cannot easily deal with, and from some points of view have little if anything to offer.

*For instance, a recent article [53] setting out the conditions for a successful expert system project specifically mentions the need for training in the LISP language. For a comparison of LISP and PROLOG see [54].

**On problems of assessing expert system development see [55], as reported in [56].

A strategic choice for development would therefore be better in the areas of construction of the knowledge base rather than in the design and implementation of another expert system shell. The construction of knowledge engineering tools to detect inconsistencies at construction time, to suggest gaps in the information supplied, to maintain check-lists of points to be covered, and to detect and suggest promising lines of enquiry in the interrogation of the human expert, are some of the types of software that could be developed. To enter such a field of work for a software manufacturer may be a promising route for some developing countries. Since experience is still limited in both developed and developing countries, it may not be as difficult to compete internationally as in some other fields. The capital equipment costs are not high, even though there will probably be a foreign exchange component.

E. General production strategy

1. Stages of software production

Traditionally, software production has been described mainly in terms of the design and implementation of computer systems for use within an organization or a complex piece of equipment. Much attention has been given to stages of such an activity. A typical classification is the following "software life cycle" ([57], p. 3):

(a) Requirements analysis and definition;

(b) System and software design;

(c) Implementation and unit testing;

(d) System testing;

(e) Operation and maintenance.

From this and similar points of view, the production of software is something which extends well beyond the actual design and writing of a programme. It calls for a number of diverse skills including human communication, personnel management, resources planning etc., as well as the traditional skills of the systems analyst or programmer. Taken together in an integrated way, the considered application of these selected skills has become the discipline called "software engineering". The rise of the discipline is a response to the growing complexity of the task. As a consequence of this development, considerable examination has taken place of the various stages of the development of a system, with rules and methodologies which can be applied to the practicalities of software production. The allied question of the costs of software development has also been explored,* especially in view of the kinds of complex systems under development, often involving large teams of staff and sometimes at different locations.

*The best-known work on this subject is [58].

However, from the point of view of the software entrepreneur, the process has not been as well analysed. The conventional analysis assumes that the idea already exists of what the software is to do. It assumes also that, on completion, the system will be used, perhaps after modification in the light of the intended user's experience of the first version. But the software entrepreneur faces two other stages: one is the initial one of deciding what to produce, and the other is the later stage of trying to sell it when completed. The task is essentially a speculative one. The intermediate stages, of design and implementation, have some similarity of characteristics, whether the software is for internal use, is being prepared to order, or is to be marketed as a package.

2. Product ideas

Ideas for new products come from an understanding of existing software and perhaps also an understanding of another specialized area. The potential software producer can be someone of a technical or professional background whose idea for software derives from an understanding of the way in which a computer can help in the work. Equally, if the potential producer has access perhaps through preparing custom software or providing computer services to some specialized company or institution, this may also provide ideas for new products.

A rough classification of ideas for new products is as follows: invention; replacement; synthesis; link-type software; captive software; local software; embedded software; and custom software.

The first type, invention, refers to the genuinely new idea for software, something that has not been tried before. While in the early days of computing much new software could have been put in this class, it is no longer easy to come up with ideas which are genuinely original and at the same time practicable. A good example is the spreadsheet programme. The first of these, VisiCalc, was certainly an invention. The hardware conditions for its invention were present in that it is a highly interactive concept: only the availability of personal computers allowed it to be successful.*

The second type, replacement, includes the presentation of existing ideas in a new form. The improvement can relate to a number of different areas, such as the speed of calculation. This is particularly important in data base software, where a "select" or "soft" can, if complex, consume considerable resources. Speed is also important for compilers and for CAD software.** Ease of use

---

*The initial success of Apple as a manufacturer has in fact been attributed to the success of VisiCalc. See [59].

**For example, in an advertisement for the Oracle data-base management system, the speed of selection, projection, and calculation is compared against that of selection in Informix, another data-base management system [60]. In an advertisement for Turbo Basic 1.1, both the speed of compilation and the speed of execution of the object programme are compared to those of QuickBasic (Microsoft) [61].

is another field where improvements can readily be made to an existing idea. This is encouraged by developments on the hardware side referred to earlier, where the user interface is enhanced by high-resolution screens which can show features such as menus, icons and windows, together with mouse pointers and speech input and out-put. An otherwise conventional package can distinguish itself and gain a competitive advantage by making full use of available features of the hardware and software. Ease of use is a wider concept. It can include the provision of extra functions within a new package to carry out operations which experience has shown are necessary but tedious to perform with the existing package. Improvement can also be in the area of removing restrictions on the size of the problem to be handled by the software, these restrictions existing because of limited imagination on the part of the original developer or hardware restrictions which have been made irrelevant by new developments.

The third type, synthesis, refers to the creation of a new class of software products by the merging of the functions of separate classes. The best example is the integrated package which can include most of the popular office software functions, such as word-processing, spreadsheet, data base, graphics and communications. As noted earlier, integration of this kind is a continuing trend, brought about very often by the increased availability of computer memory. Very often, the process of synthesis exploits the innovative character of earlier products. What was a new idea, and was sold separately, becomes a feature of the later package. There is, however, a sense in which the combination of existing ideas also represents a new idea. For this reason the search for new product ideas has to include consideration of existing ones, with special attention being paid to the possible merging of ideas from less obviously linked areas.

Link-type products represent a promising field for software development, but it is one which changes very quickly. By a link-type product is meant one which overcomes improved communication between systems of hardware and software. The definition thus includes software to carry out such tasks as converting a file from one format to another, or translating some code from one language to another. It could also include software to allow, for instance, a computer to use a printer which had not been designed for this type of computer. At first sight, the possibilities for this sort of product might be thought to be limited in the longer term by the trend towards standardization, but this is not so. First, universal standards are still years away; secondly, even the best standard is made obsolete by techno- logical change; and thirdly, the very push towards standards creates a need for software products which allow non-standard hardware and software to communicate.

Captive software is the name used to describe software products such as "add-ons", which are software providing extra features or enhancements to successful products. It can include also other products which are intended to improve the performance of successful products, or, indeed, are intended to monitor the performance of complex software in a way which will allow improvements in its use.* All these share a common characteristic, that they depend on the success of another software product. Their potential market size is determined by the number of copies sold of the other product. More importantly, it is very vulnerable. For instance, the producers of the original product may decide to bring out a new version which incorporates most of the features of the add-on products. Again, a new version may be incompatible with the existing version of the add-on. This may not be intentional on the part of the producers of the main software, but it can nevertheless create considerable difficulties for those whose products are supposed to work with it. For this reason such software products can be called captive: they are wholly dependent on the success of another. In that sense they are analogous to software products which are almost always either hardwareor operating-system-dependent, or both, but the difference lies in their limited functionality. What they do for the user can be quite restricted, and the need for the function as well as the product can be eliminated at any time by developments outside their control.

Local software means the meeting of a particular national practice by appropriate adaptation of existing software or software ideas. This can amount merely to changes in the language of the command structure or user interface of a package. If, however, the language is very different, there will have to be more elaborate changes. Local business or governmental procedures in the form of accounting, taxation etc. represent the most common origins of local software. Although local software is usually produced in the country concerned, or else carried out for instance at the European headquarters of a software package producer of a developed country, it does not have to be so. A developing country software company that produces a standard accounting package to meet the special needs of the country has also mastered many of the skills needed to do the same for another developing country.

Embedded software is that contained in a machine vehicle, or piece of equipment. As such it is usually developed by or on behalf of the manufacturer, and is similar to the final category, custom software, in that the aspiring software entrepreneur has to produce this product to order and is essentially a subcontractor. While the product itself is not marketed by the software producer, there is nevertheless, especially for the new producer, a good deal of marketing involved, since it is necessary to convince the client that the software company has the skills and experience to carry out the task correctly and on time. The difference between embedded software and custom software is partly in the markets: manufacturers of capital goods and consumer durables are the typical

---

*For instance, the very successful data base software for personal computers, dBase III, has engendered many other products [62]. The operating system itself, MS-DOS, has done the same, for instance with "shells" to improve the user interface (see [63], as reported in [64]). On the mainframe side, a large number of performance monitors are available; one source gives 43 for IBM systems [65].

users of embedded systems, while custom software could be written for anyone. Embedded software is used by microprocessors or micro-controllers built into the equipment. It is usually very time-dependent and written in assembler or in a special control language (increasingly, Ada is being used). Embedded systems construction may also in many cases require a much more detailed knowledge of hardware considerations.

3. Choice of market

Clearly, the possible market will be partly determined by the idea for the product. The idea may be in practice specific to a particular application sector, a particular country or a particular type of computer or operating system. To have as large a potential market as possible is desirable, but in the case of software production it is particularly so because of the low marginal costs involved in expanded production.

In terms of large potential markets, consideration must be given not only to the installed base (the number of relevant computers in use) but the type (mainframe, minicomputer or personal computer). The price at which the product can be sold varies accordingly, inversely with the number installed: thus personal computer software, unless highly specialized, can hardly be sold for more than about $600. Minicomputer and mainframe software is considerably more expensive, and in fact is often rented, thus giving the producer a certain control over future revenues.* The question of pricing in general is also affected by technological change: the shortening of product life cycles in the software field means that pay-back periods have also to be shorter if they are to exist at all.

A full understanding of the selected market is necessary, in particular the motivations of the potential software purchasers, which vary widely depending on the context in which their computing work is carried out. For instance, the IBM mainframe installations constitute a large and stable market (with respect to minicomputers, a similar judgement can be made). However, the mainframe installations also constitute a market which has the following characteristics:

(a) It is concerned as much or even more with quality and service as with price;

(b) It has already been targeted by many others;

(c) It is in fact fragmented, since so many software products have become "standards" within this typical environment. It is not enough to be compatible with MVS. There is other system software

---

*It is not an absolute control, because to raise rents too much would open the door to a rival producer. But the inconvenience of transferring to another software system would in any case deter many from changing.

(in teleprocessing, for instance) which has to be taken into account. Sophisticated heavily marketed products such as data-base management systems have specialized formats and imply certain ways of working. Can the new product fit in with these?

(d) It is an evolutionary market, not very much open to significant departures from established ways of doing things. This is principally because of the need to keep established computer systems operating. These have often entailed considerable investment costs and they provide information flows which cannot be interrupted. For this reason, any new product has either to complement existing systems or else provide a feasible way to replace the existing system in an orderly manner. As noted earlier, the computer manufacturers have to provide a stable and secure environment for the purchaser. The user has to feel that the equipment bought is not likely to become obsolete quickly. In fact, it usually does become obsolete quickly, in the sense that something both better and cheaper will appear on the market soon after the purchase has been made. But it can be said that the purchasers grudgingly accept this, if their individual expectations are met, if the available equipment is within their budget, if it will do the job it is intended for, and, most importantly if it can either be physically upgraded (which is better) or else will use the kinds of software that can also be used on the next generation of the computer in question.

This last requirement means that the computer manufacturer has to provide an "upgrade path". This can take several forms, but the simplest is a set of programmes that convert software written for the old computer to software that will run on the new. Such a conversion is not always necessary since the hardware may have been designed to be "upward compatible". However, on other occasions it may be necessary to abandon such compatibility in order to exploit fully the possibilities of the new hardware design or components. It is in these cases that the provision of an upgrade path is needed. Such a path can also include the provision of hardware modifications, and the prevalence and cheapness of integrated circuits in many items of equipment makes this a relatively simple task, since the replacement of one integrated circuit by another is a relatively simple matter. However the provision of software that converts old systems to allow them to operate in the new environment is one of the most usual ways in which an upgrade path is made available.

There are cases, however, in which an upgrade path is not provided by the manufacturer. This can result from the purely practical calculation that the returns will not justify the development effort, but it can also be that the benefits of the new technology are sufficiently obvious and significant for the user to simply abandon his old systems, running them in parallel with the new ones only until the latter have proved their reliability.

In other cases there may be a "generic" shift in the technology. An example is the movement from 8-bit to 16-bit microprocessors, where the earlier generation (8080, Z80 and 6800) was not compatible with the later (8086, Z8000, 68000) in terms of tne instructions understood. Individual manufacturers may have provided upgrade paths for their own series of microprocessors, but many microcomputer manufacturers took the opportunity of a technological shift to change their sources of microprocessors. Third-party software companies, as well as chip manufacturers, offered products, cross-assemblers, which allowed the instructions of one microprocessor to be translated into those of another. Thus, technological change created a gap which was filled by third parties. Many other gaps necessarily arise, as long as the technology continues to develop, and such gaps always represent opportunities for third-party manufacturers to provide a product which can fill the gap and allow two previously incompatible pieces of equipment to communicate with one another.

In considering markets, therefore, it is important to reiterate that technological change creates markets in two distinct ways. Firstly, through direct progress and the provision of new hardware it encourages new ways of doing things and, accordingly, new software requirements for software products which exploit the possibilities of the new hardware. But there is a second type of market, one created by the inconsistencies and discontinuities of technological change. Progress continues in many directions, and led by many different companies in several countries. To reap the benefits of more than one advance simultaneously may mean, for a new product developer, a difficult problem of integration: how to fit together a sensor, a processor and a memory management unit in such a way that it will be of interest to users of another product that happens to be highly popular but with which no suitable interface exists. Integration, harmonization, communication are all goals of the movement towards hardware and software standards. But in the absence of uniform acceptance of standards, and in the face of the continued assault on standards represented by accelerating technological change, the need persists for products to fill the gaps between the advances on different fronts. The link-type product idea is only one manifestation of technological change, which increasingly determines the whole market environment.*

It was pointed out above that simply to follow what appeared to be the largest or "standard" configuration of equipment may not necessarily be the best approach. On the other hand to follow a more specialized market carries its own risks. One of these is that

---

*Filling gaps, as a strategy, involves an appreciation of many aspects of hardware and software. It may also involve working with hardware directly to such an extent that the operation may seem more like hardware manufacturing. But in fact there is no physical transformation of the materials, apart from the linking of chips and connectors together. The exact pattern of linkage may be unique but it uses printed circuit board techniques that are well understood. The real contribution and the source of value added comes from the software, which can be embodied in read-only memories (ROMs), erasable programmable read-only memories (EPROMs), electrically erasable programmable read-only memories (EEPROMs) and application-specific integrated circuits (ASICs).

the market may be about to be eroded, or may be wiped out by the emergence of some new technology or a rival product. Here the development time is crucial in a way that is not necessarily true for products directed towards a more stable or conventional market. In the latter case, a product may still find buyers whether it appears in 12 or in 18 months. In a more specialized market, it may contract rapidly as the technology changes and the total number of potential purchasers can diminish rapidly. This is still the case even if the number of computers of a particular kind in use remains the same. Although none are immediately abandoned on the announcement of the availability of some superior product, decisions are nevertheless taken on future purchase. Once such decisions are made, it is then only for very compelling reasons that any software product would be bought for the existing computer. Even though it may remain in use for another year or even more, the software market is disappearing rapidly.

Being too near the frontier of hardware developments is equally risky, since the new development may fail to attract a wide market and the work that goes into building software for it, together with the consequent investment, may be lost. However, many promising niches exist at this frontier. New developments in the mainstream of hardware are also important and here the established manufacturer of software is often at an advantage over new entrants. This occurs particularly in the case of, for instance, microcomputers, whose manufacturers are anxious to launch the machine on the market with as wide a software base, or selection, as possible. This means that they will make available to software companies a prototype version of the new computer, or at least disclose to them the details of the hardware and operating system so as to allow the software manufacturer both to have a product ready for market launch simultaneously with the launch of the computer itself, and also to enhance the attractiveness of the computer to potential purchasers through the provision of software that exploits any special feature of the new hardware design. The companies selected for this favoured treatment are typically those that have produced successful packages in the past, since too wide a distribution would discourage some of the software manufacturers, who would feel that their competitive advantage was being lost, and might also result in disclosure of any still secret aspects of the hardware design.

Such a practice makes it more difficult for new entrants to compete. It favours existing successful software companies. But the practice was not developed for the sake of these companies, but rather by the hardware manufacturers to ensure success for their products. Therefore, they remain to some extent open to broadening their range of favoured software companies, provided they are satisfied that the company entrusted with details of their forthcoming product is capable of producing something worthwhile in the way of software for it. If can therefore become very important to cultivate selected hardware manufacturers, to demonstrate the software capabilities of the new company and to convince the hardware developers that the connection is worth developing. Such a task may not be an easy one: it can require in some cases a physical presence in the same country as the hardware manufacturer or at least very frequent visits to it, and for a small company in a developing country this may be an overambitious undertaking, given the foreign exchange costs of establishing a presence in North America or Western Europe.

However, some such contacts are essential for any software production strategy targeted towards the leading edge of technology. To be familiar enough with present and future developments to be able to identify an emerging software product opportunity is not easy when the research and development activity is far away and sometimes hidden for commercial reasons. But there are many relatively inexpensive methods of monitoring technological progress in this field. Journals in electronics and computing are often inexpensive because of the high quantities of advertising that they carry. Professional societies are usually very keen to expand their international links and often have a proliferation of specialized interest groups which act as a clearing-house for information (and often standards) on particular hardware or software topics. Again, hardware manufacturers often provide a great deal of printed material on their own products or on technologies which they use. This material is often free or else costs very little. Attendence at large trade fairs, such as CeBIT or Comdex, is a relatively easy way of becoming familiar with a complex and rapidly changing market.*

Co-operative arrangements for exchange of scientific and technological information are found both within the United Nations system and as part of other intergovernmental arrangements. UNIDO has actively promoted information exchange both through the Technology Information Exchange System and more specifically in the field of informatics technologies through schemes such as the REMLAC regional micro-electronics network for Latin America. The fledgling software company can do worse than investigate official channels in its own country, to discover what information systems are accessible as a consequence of international agreements.

The question of software protection should also be mentioned. The subject is a complex one since it includes technical as well as legal considerations. The investment made by the producer has to be protected through some system or systems which will prevent unauthorized copying of the software. This can lead to a loss of potential revenues, and means to guard against it range from physical protection of the software diskette to the application of patent and copyright law.**

---

*Representation at these fairs of the new software company is another matter and often an expensive one. The average cost of the space for a stand at CeBIT is 200 deutsche mark (DM) per square metre, and to this must be added the construction of the stand and many other associated costs. The total expenditures of Siemens in 1989 was DM 8 million. See [66].

**See [67]. On the terms and conditions of software sales, see [68].

4. Investment costs

The investment costs associated with software production include the provision of hardware and software equipment. Clearly there will be a need for computer hardware, the same as or better than that for which the software will be written, although development hardware which can simulate the largest hardware may also be a possibility in a few cases. Inefficient hardware or limited access to it does not make it impossible to produce marketable software but it adds to the development time. This might not appear to be so much of a problem if local personnel costs are low, but if the product under development is in any way sensitive to technological change, then development time can be decisive in determining the success of the product. Too long a wait may make it obsolete, or the problem it is to solve has been taken care of by someone else.

Investment in software is at least as important for this reason, and for others. The choice of the correct tools will not only reduce the development line but will also make the final product more attractive and reliable. Software products are available which are designed to assist in all stages of the production of software, including design, production, documentation and maintenance. The products include the following:* design tools, which allow the conceptual structure of the software system to be analyzed; optimizing compilers, which produce efficient object codes by analyzing the source code for redundancies; fourth-generation languages and application generators, which may also have a number of other tools as features; screen generators; many forms of debuggers, which check the source code for errors; execution flow summarizers; file comparators; and translators.

Many of the features found in such software tools are also to be found in the new generation of software tools known as computer-aided software engineering (CASE). These include what are called programmers work-benches or analyst work-benches. The more comprehensive type is called an integrated projects support environment (IPSE), capable of controlling large software projects and ensuring consistency in the work of the different people involved and generating comprehensive documentation for the product.

These products vary in capabilities and in price. Their potential contribution to the success of the final product can, however, be very great, and productivity improvements of from 20 to 30 per cent have been cited of IPSEs [71]. The cost of such tools may therefore be a centrally significant feature of any detailed feasibility study for a software product, and the savings in labour inputs and project time, as noted, must be taken into account. This is so although estimating the work involved is difficult. Most cost estimate methods for software production require the number of lines

---

*See [8] and [57]. See also [69], as reported in [70], for some examples of COBOL tools for an IBM environment.

of code as a starting-point.* It can also be important to follow design procedures which allow generalized parts of the software to be used again in subsequent products ("re-usable code").

The second main advantage of such tools, that they greatly reduce the potential for errors in the final product, will have an impact on the cost structure at a later stage. A faulty product will not only give the new producers an unfavourable reputation,** it may also lead in some cases to liability for consequential damages. As a result, there has been a growth in independent companies which for a fee will test a software product for errors before it is marketed. Their customers include very large and successful software producers [75].

---

*See [72], as reported in [73], and also [58].

**It has been suggested, however, that the maxim "any publicity is good publicity" applies here, but this seems to be more in connection with large companies who can supply updates at relatively little cost to registered users and thus get a good reputation for quality control. See [74].

References

1. Organisation for Economic Co-operation and Development, Software: An Emerging Industry, Information Computer Communications Policy No. 9 (Paris, 1985).

2. International Standard Industrial Classification of all Economic Activities, Statistical Papers, series M, No. 4, Rev. 2 (United Nations publication, Sales No. E.68.XVII.8).

3. Standard International Trade Classification, Statistical Papers, series M, No. 34, Rev.2 (United Nations publication, Sales No. E.75.XVII.6).

4. Organisation for Economic Co-operation and Development, Working Group on Accounting Standards, "Accounting treatment of software", Working Documents, No. 1 (Paris, 1986).

5. Robert Schwarz, "Software industry development in the Third World: Policy guidelines, institutional options, and constraints", World Development, vol. 15, No. 10/11 (1987), pp. 1249-1267.

6. Lydia Arossa, Internationalization of Software and Computer services. ICCP/OECD Report to be published in 1988.

7. 1986 US Industrial Outlook (Washington, D.C., United States Department of Commerce, January 1986).

8. B. W. Boehm, "Improving software productivity", IEEE Computer, September 1987.

9. "Survey: software packages for business", Financial Times, 1 May 1985.

10. "Survey: computer services", Financial Times, 15 October 1987.

11. Organisation for Economic Co-operation and Development, Standards in Information and Communications Technology (Paris, 1987).

12. United Nations Economic Commission for Europe, Software for Industrial Automation (United Nations publication, Sales No. E.87.II.E.19).

13. P. Jackson, "Add-on flood keeps the tide flowing into Lotus channels", Computer Weekly, 9 April 1987.

14. Charles E. Gimaric, and Vjelko M. Milutinovic, "A survey of RISC processors and computers of the mid-1980s", IEEE Computer, September 1987, pp. 59-69.

15. "Forget the hype, this is the real thing", Computer Weekly, 23 April 1987.

16. Official Journal of the European Communities, No. C 46/19, 18 February 1988.

17. Robert Bernhard, "Computer architecture: computing at the speed limit", in Next-generation Computers, Edward A. Torrero, ed. (New York, Institute for Electrical and Electronics Engineers, 1985).

18. Kenneth Labich, "The shootout in supercomputers", Fortune, vol. 117, No. 5 (29 February 1988), pp. 37-40.

19. "Titans engage in battle over tomorrow's systems", Computer Weekly, 26 May 1988.

20. J. Weber, "A window into the brain", Personal Computer World, March 1988.

21. Jeff Conklin, "Hypertext: an introduction and survey", IEEE Computer, September 1987.

22. Theodor H. Nelson, "Managing immense storage", Byte, January 1988.

23. IMC Journal, vol. 24, No. 4 (1988).

24. "Nanobytes", Byte, August 1988.

25. United Nations Industrial Development Organization, "Software production: organization and modalities" (UNIDO/IPCT.63, 17 May 1988).

26. J. Martin, An End-user's Guide to Data Base (Englewood Cliffs, New Jersey, Prentice-Hall, 1981).

27. "A tarnished UNIX fails to lift AT&T", Computer Weekly, 16 July 1987.

28. J. Isaak, "Designing a standard", Computer World: Focus, vol. 20, 20 August 1986.

29. "Group of ten challenges IBM", New Scientist, 27 November 1986.

30. "Just wild about Posix", Computer Weekly, 29 January 1987.

31. "Software: Why AT&T and Sun hooked up", Electronics, 29 October 1987.

32. IEEE Computer, July 1988.

33. Computer Weekly, 26 November 1987.

34. R. Malloy, "First impressions: IBM's OS/2 extended edition", Byte, July 1988.

35. R. Fertig, "IBM sets its sight on greater software revenues", Computer Weekly, 27 March 1986.

36. R. Fertig, "What IBM has in store for users", Computer Weekly, 11 June 1987.

37. Byte, August 1988.

38. B. W. Kernighan and D. M. Ritchie, The C Programming Language (Englewood Cliffs, New Jersey, Prentice-Hall, 1978).

39. I. C. Pyle, The Ada Programming Language (Englewood Cliffs, New Jersey, Prentice-Hall, 1981).

40. O.-J. Dahl, E. W. Dijkstra and C.A.R. Hoare, Structured Programming (London, Academic Press, 1972).

41. H. Mills, "Structured programming: retrospect and prospects", IEEE Software, vol. 3, No. 6 (November 1986).

42. E. F. Codd, "A relational model of data for large shared data banks", Communications of the ACM, vol. 13, No. 6 (June 1970).

43. Computer Weekly, 8 October 1987.

44. "Inside technology", Electronics, 29 October 1987.

45. "Feature: design automation", Electronics Weekly, 6 July 1988.

46. P. Jackson, Introduction to Expert Systems (Reading, Massachusetts, Addison-Wesley, 1986).

47. Computer Weekly, 9 January 1986.

48. J. Lamb, "Translation systems: bridging the language gap", Datamation, vol. 32, No. 1 (1 January 1986).

49. United Nations Industrial Development Organization, "Applications of pattern recognition and image processing to industrial problems in developing countries" (UNIDO/IS.609, 14 February 1986).

50. United Nations Industrial Development Organization, "Expert systems: prospects for developing countries" (UNIDO/IPCT.41 (SPEC.), 23 September 1987).

51. E. Gahan, "The impact of expert systems", Industry and Development, No. 23 (United Nations Industrial Development Organization publication, Sales No. E.88.III.E.3).

52. "AI woos MIS anew", Computerworld: Focus, 1 June 1988.

53. J. M. Cupello and D. J. Mishelevich, "Managing prototype knowledge/expert system projects", Communications of the ACM, vol. 31, May 1988.

54. Walter F. Tichy, "What can software engineers learn from artificial intelligence?", IEEE Computer, November 1987.

55. Wayne J. Socha, "Problems in auditing expert system development", EDPACS, vol. 9, March 1988.

56. Data Processing Digest, No. 6, 1988.

57. I. Sommerville, Software Engineering, 2nd ed. (Reading, Massachusetts, Addison-Wesley, 1985).

58. B. W. Boehm, Software Engineering Economics (Englewood Cliffs, New Jersey, Prentice-Hall, 1981).

59. "Survey: personal computers", Financial Times, 14 September 1988.

60. UNIX/World, December 1987.

61. Personal Computer World, September 1988.

62. "dBase add-one", Personal Computer World, September 1988.

63. "Eight DOS shells that make computing easier", Infoworld, vol. 9, 19 January 1987.

64. Data Processing Digest, No. 3, 1987.

65. Xephon Technology Transfer, Xephon Publications, Autumn 1987.

66. "Wieviel ausgegeben wird, um Geld zu verdienen", Computerwelt, No. 5, 11 March 1988.

67. G. G. Davis, Software Protection: Practical and Legal Steps to Protect and Market Computer Programs (New York, Van Nostrand Reinhold, 1985).

68. United Nations Industrial Development Organization, "Trends in commercialization of software in developing countries" (UNIDO/IS.574).

69. David Roman, "Classifying maintenance tools", Computer Decisions, vol. 18, 30 June 1986.

70. Data Processing Digest, No. 11, 1986.

71. "IPSEs start to come into their own", Computer Weekly, 4 February 1988.

72. Howard A. Rubin, "A comparison of software cost estimation tools", System Development, vol. 7, May 1987.

73. Data Processing Digest, No. 8, 1987.

74. "What's bugging you", Practical Computing, July 1986.

75. "Software makers battle the bugs", Fortune, 17 February 1986.

 18255

BUSINESS FAILURE IN GREEK MANUFATURING INDUSTRY: SOME EVIDENCE FOR POLICY

George H. Katsos and Joseph N. Lekakis*

Introduction

The important issue of business failure arouses the concern of investors, bankers, managers and policy-makers. In literature on the subject an attempt has been made to determine the degree of influence exerted by both intra-firm and aggregate external factors on failure rates. Intra-firm factors are associated with the type and quality of management, which is responsible for business strategy and progress, while external factors include macro-economic conditions and policies adopted outside the business community.

In a study of intra-firm factors, J. Argenti ([1], p. 123) reports that bad management is almost universally recognized as the prime cause of business failure, and identifies the following six main structural defects of management: one-man rule; non-participation by the board; existence of an unbalanced top team; lack of management depth; poorly organized finances; and the practice of combining the roles of chairman of the board and chief executive officer.

With regard to external influences, the same author ([1], pp. 128-130) suggests that price controls, strict environmental regulations, and other technological, political, economic and social factors are significant determinants of failure. In a more recent systematic study of external influences, C. Cumming and K. Saini [2] built a single equation model based on the underlying assumption that the factors determining failure at the firm level can also be used to explain the phenomenon in the aggregate. In their model, they regress the number of bankruptcies in Japan and the United Kingdom of Great Britain and Northern Ireland against the price level and aggregate demand, costs and interest payments. The major conclusion drawn from their work is that bankruptcy growth rates are explained to a large degree by debt service costs and individual components of aggregate demand. Altman ([3], pp. 83-98) chose a different set of independent variables to produce findings indicating that marginal firms in the United States tend to fail much faster when there is a decline in economic growth, stock market performance, money supply growth, and business formation. He also notes that the literature on the issue of aggregate external factor influence on business failure is limited ([3], p. 84). Finally, in a study of the effects of federal credit policies and aggregate business failure in the United States, Archibald and Baker [4] conclude that the non-market allocation of

*Research Associates, Centre of Planning and Economic Research, 22 Hippokratous Street, 106 80 Athens, Greece. The views expressed in this paper are solely those of the authors.

credit to firms that have a lesser probability of survival contribute to higher failures. In addition, federal loan guarantee programmes merely postpone failures which then eventually grow into larger failures.

Macro-economic policy choices acting as aggregate external factors may have profound effects on business performance, especially in countries where government intervention is frequent and decisive. The role of trade, taxation and financial and other policies has often been emphasized in the literature, with the main focus on paths to development [5] because national efforts since the Second World War have been concentrated on the creation and expansion of industries.

The experience of failing manufacturing firms in Greece is considered in this paper within a dual microand macro-economic framework, thereby contributing to the existing scarce literature on the subject, and useful policy conclusions are drawn. The few sectoral studies of the Greek economy have touched upon some of the causes of failure in the manufacturing industry. Inadequate management, characterized by a lack of innovation and a failure to deal efficiently with the task environment, has been suggested in a study by Tsoris [6]. Giannitsis and Vaitsos [7] have also pointed out the lack of technological innovation, but without detailed consideration of the causes. In another work the thesis of a lack of technological innovation is rejected ([8], p. 39). In a few studies it has been observed that many failing firms have had a high debt-to-equity ratio [9]. The basis for that observation, which reflected the way in which manufacturing industry was originally created through abundant credit, business choices and government policies, will become clear in this paper. A few other studies have revealed that, in order to finance new investment projects, firms have had to borrow in the short term ([10], [11] and [6]). None of the studies, however, systematically addresses the problem of failure in the context of policy-making, on the basis of information received directly from firms.

The approach followed in this paper is guided by the belief that micro-level data are of invaluable significance for the analysis of business failure, and that evidence associated with both intra-firm and external causes of failure can be incorporated into the decision-making process. Data have been drawn from various sources, including the National Statistical Service of Greece, the Bank of Greece, ICAP (a private consulting company), the Union of Greek Industrialists and personal interviews with manufacturing business officials. Before business failure is considered from an analytical perspective, data on manufacturing growth from the end of the Second World War up to 1974 are presented in section A. The year 1974 is chosen as a dividing line for the following three reasons: it was the year in which the pressures produced by the first oil crisis began to spread their effects through the domestic economy; it saw the departure of the military government; and it gave rise to new challenges for economic policy in Greece. The experience of failure in Greek manufacturing industry is summarized in section B. A discriminant analysis model is developed in section C using variables assembled from company financial statements, in order to trace the relative importance of the factors determining business failure. The results of a set of tests for effective management in solvent and insolvent firms are presented in section D. The relative impor- tance of aggregate external factors as causes of failure in the Greek manufacturing sector are explored in section E, with emphasis on cost, price and demand conditions. Finally, in section F, the Greek experience is summarized and policy conclusions are drawn.

A. Growth of the manufacturing sector in Greece

The manufacturing sector in Greece started to grow at a brisk pace immediately after the Second World War as a result of the general reconstruction effort in transports, agriculture and energy. Planning for manufacturing growth effectively began in 1953, with emphasis on energy. During the same period, growth was mainly centred on certain development projects in the fertilizer, aluminium, magnesium, nickel, soda, sugar, steel, oil refining and ship repairing industries [12]. Invisible receipts and the inflow of private foreign capital, supplemented by the aid programme of the United States of America, were identified as the main sources of foreign exchange needed to import machinery and equipment. By those means, a 6 per cent growth of gross national product (GNP) per annum could be sustained ([13], p. 2).

The absence of private capital markets compelled policy-makers to use bank credit as the sole means of financing manufacturing investment projects. Development of such markets was inhibited because large labour migration from rural to urban centres, signalling the transformation of Greek society from a peasant to an urban one, made investment in housing a major priority ([14], p. 70). Credit to manufacturing increased by approximately 32 per cent per annum throughout the period from 1953 to 1973, during which Greek manufacturing industries achieved remarkable growth in real terms (9.5 per cent per annum), following a 7 per cent annual increase in aggregate demand.

Greece did not apply strong import restrictions as a means of spurring manufacturing growth for various reasons. Beyond equipment and raw materials, imports of a variety of foreign-made products were essential for further growth. More essential, however, was the choice of the country to become an open economy. As early as 1953, almost all quantitative restrictions on imports were lifted, and export subsidies, which had previously amounted to 100 per cent of f.o.b. prices, were abolished. Further, in 1961, aiming at a higher export performance, Greece joined the European Economic Community (EEC) as an associate member. Goals set by the five-year development plan for the period from 1968 to 1972 [15] favoured the creation of large firms to reap economies of scale. Small firms, however, have been found to be more efficient, and hence less vulnerable to economic downturns [9]. In 1970 the authorities changed course and decided to subsidize exports up to a maximum of 25 per cent of f.o.b. prices. Exports increased rapidly between 1970 and 1974.

B. The failure of manufacturing firms

Immediately after assuming power, the civilian Government reshuffled its expenditures. Fearing a possible conflict with neighbouring Turkey (following the military action of 1974 in Cyprus), the Government increased national defence spending. In addition, the first oil crisis forced it to allocate more funds to energy imports. As a result, there was a significant cutback in resources available to manufacturing, including public investment, various subsidies and grant programmes. Exports subsidies, specifically, were reduced and gradually abolished, reflecting the above-mentioned factors and the obligation of the Government to pave the way towards full membership of the EEC.

The inflationary spiral which appeared at that time reflected the relative scarcity of resources. Unit production costs in manufacturing industry began to increase rapidly. Simultaneously, in an attempt to avoid the negative impact of continuing inflation on the volume of private deposits, the Government raised nominal interest rates and, consequently, the cost of lending. Increased production costs, rising competition from newly industrializing countries* and the European recession of 1976 to 1980 had adverse effects on business profitability. The expectations of investors soon followed the downward profitability trend and the manufacturing index decreased rapidly. Even though the stock market is not a significant source of business financing, the end-result was that real growth rates in manufacturing investment approached zero, and in certain periods even became negative.

In 1977 the Government openly spoke about the existence of "ailing" or "problem" (the more popular term) firms that were having difficulty in meeting their short-term obligations. In 1983 it announced a plan to bail out such firms, in an attempt to avoid the increased unemployment that would arise from the closure of manufacturing plants. Problem firms were either bailed out or declared bankrupt. The immediate result of supporting a firm under the plan was the installation of new management chosen by the Government. The manufacturing branches in which business failures were concentrated included beverages, tobacco, textiles, clothing, paper and paper products, machinery and tools.

C. Evidence from financial ratios

The analysis of financial ratios provides valuable insight into the causes of business failure. The aim is to determine the relative weight of variables that may be used to explain failure or success at the firm level. To develop an appropriate framework, a five-variable discriminant analysis model was applied to a sample of 104 firms, half of which were ready to close down in 1982, while the other half had no serious problems. Each group was matched

---

*The term "NIC" is used extensively to describe developing economies, be they countries, provinces or areas, where there has been particularly rapid industrial growth.

with the other in terms of size and manufactured products. Evidence from this part of the analysis is discussed later in the context of both intra-firm and aggregate external influences on business performance. The five variables selected are as follows:

X_1 = Net working capital/total assets

X_2 = Retained earnings/total assets

X_3 = Earnings before interest and taxes/total assets

X_4 = Equity and retained earnings/total liabilities

X_5 = Sales/total assets

The model is comparable to that constructed by Altman, with one exception. Since the stock of many of the firms is not traded, the definition of the variable X_4, in this analysis is a proxy to that used by Altman ([3], pp. 102-125), defined as the market value of equity divided by total liabilities.

The estimated Z function has the following form:

$$Z = -0.523 + 4.56 X_1 + 0.27 X_2 + 0.01 X_3 + 0.46 X_4 + 0.75 X_5$$

with the scale vector being X_1, X_3, X_4, X_5, X_2, the sequence showing the relative importance of each variable in explaining business failure or success. The model classified firms as failed or not failed with a probability of 0.86, with lower and upper limits of the Z values being -0.27 and 0.28 respectively. The primary result, as expected, is that failure is attributed to low cash-flow capabilities of the firms.

D. Intra-firm causes of failure

In any firm, cash-flow problems are primarily the responsibility of the management. Three tests of effective management were conducted. The first test dealt with gearing behaviour, the second with the firm's ability to respond to changing competitive conditions, and the third with the structure of management.

In the sample of problem firms, it was found that gearing behaviour increased from 71.9 in 1973 to 87.5 per cent in 1980, compared with a slight decrease from 65.6 to 63 per cent in non-problem firms during the same period. Hence total liabilities had a profound impact on the ability of problem firms to meet their short-term obligations, that is, variable X_4 in the model influences X_1.

The two sample groups responded in diametrically opposite ways to the rising competition of the 1970s, which originated in the newly industrializing countries and the European recession of the period from 1976 to 1980. Instead of retrenching their export activities, as did their healthy counterparts, problem firms acted rather irrationally by undertaking additional export-oriented investment projects. New projects were financed with borrowed capital at a time when interest rates could be expected to rise in response to preceding inflationary pressures. In problem firms, exports as a percentage of sales changed from 26.8 per cent in 1976 to 31.8 per cent in 1980, in contrast to healthy firms, exports of which dropped from 42.8 to 30 per cent in those same years.

The top management structure was tested in 34 large firms employing approximately 20,000 persons, which represents approximately three quarters of the total labour employed in 150 firms that were ready to close down in 1982. In 27 of the firms in the test sample at least two board members belonged to the same family; in nine firms more than four board members came from the same family; and in 12 firms housewives were also registered as board members. In 14 firms two or more members had the same profession, while in an almost equal number of firms (13) the chairman of the board was also the chief executive officer. These features are perfectly consistent with those identified by Argenti [1] for bad management, namely one-man rule and the existence of an unbalanced team at the top. The accounting information supplied to management in the test firms was adequate, according to opinions expressed by experts employed by those firms.

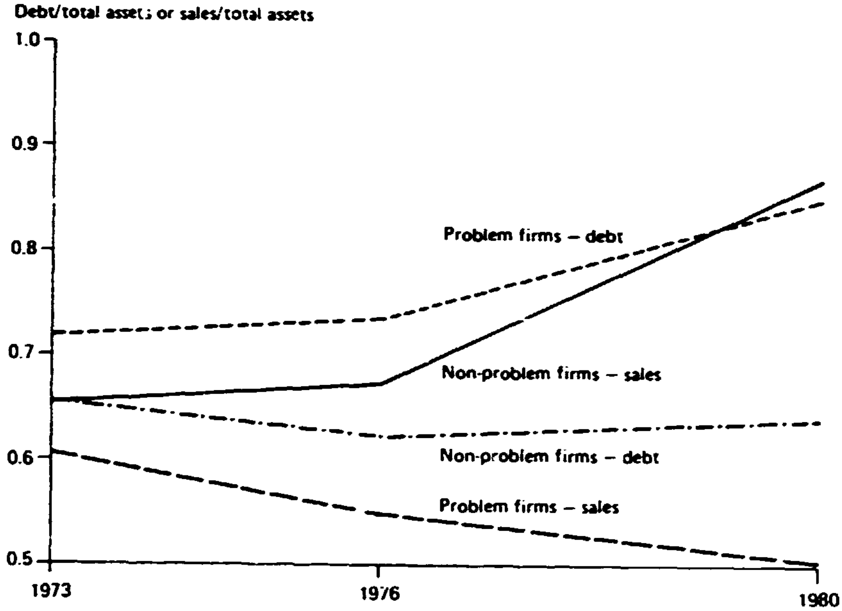
E. Aggregate external influences and business failure

Cash-flow problems in any open economy such as that of Greece are almost by definition a result either of a lack of adequate demand that has a negative impact on sales and business profits, or of increased costs combined with fixed or decreasing prices that lower profit margins. Those variables provide a broad framework that encompasses the Cumming-Saini model as well as most of what Altman regards as external causes of business failure. The factors that affect the two variables therefore need to be delineated.

As a basis for the analysis, it should be noted that in the scale vector obtained using the discriminant analysis model, variable X_4, which is cost-related, ranks higher than variable X_5, which is demand-related. A graphical representation of debt and sales profiles over time illustrates the different trends. The figure shows debt and sales ratios of problem and non-problem firms for the years 1973, 1978, and 1980.

In the figure, problem firms appear to combine falling sales with rising debt, in contrast to their healthy counterparts whose sales ratios increased while their debt profile showed no significant change. This does not mean that the sales of problem firms were in fact decreasing. It was the increasing value of the denominator (total assets) caused by the accumulation of new capital assets that made their sales ratios follow a downward pattern. On the basis of the sample data, the average annual growth rate of fixed asset accumulation by problem firms during the period from 1973 to 1982 was almost 12 per cent, as compared with only around 5 per cent for non-problem firms.

DEBT AND SALES RATIOS OF PROBLEM AND NON-PROBLEM FIRMS
(1973–1980)



1. Demand conditions

The assessment of demand conditions as determinants of reduced business performance requires tracing aggregate demand components over time, in order to identify possible constraints that had an effect on the emergence of problem firms. For the period after 1974, table 1 shows individual demand components comprising exports, both competitive and non-competitive imports, and domestically produced and demanded manufactured goods.

As reflected in table 1, exports of manufactures to European member countries of the Organisation for Economic Co-operation and Development (OECD) grew at rates which did not differ from those recorded by other member countries. Goods produced by problem industries appear to have experienced even higher export growth rates. Consequently, low exports did not result in a lack of aggregate demand.

Imports grew at rates more or less equal to those for other OECD member countries. However, foreign goods similar to those manufactured by problem firms were imported at much higher rates than those of total imports. The high growth rates of demand for imports reveal the existence of significant import penetration, the index of which rose from 0.325 to 0.360. This implies that domestically produced goods had become less competitive because of events or policies that had a negative effect on relative prices.

Table 1. Aggregate demand components, 1974-1982
(Annual rates of growth)

Demand component	In total manufacturing a/	In problem industries b/
Imports		
Greece	8.0	18.5
European members of OECD	9.1	-
Exports		
Greece	9.9	15.5
European members of OECD	8.8	-
Domestic demand for domestically produced goods c/	5.8	3.5

Source: Organisation for Economic Co-operation and Development, Foreign Trade by Commodities (Paris, 1986), vol. 1.

a/ Categories 5, 6, 7 and 8 of the Standard International Trade Classification.

b/ Industries in which problem firms appeared in greater proportions.

c/ Calculations based on data from the National Statistical Service of Greece.

All of the above-mentioned factors lead to the conclusion that demand was not declining. Evidence from micro-level data must therefore be incorporated to assess the adequacy of aggregate demand. Specifically, an interpretation of sales profiles for the problem and non-problem firms referred to in the figure is needed. The first-best hypothesis is that unlike problem firms, non-problem firms had increasing sales because of their ability to become more competitive by international standards.

2. Cost conditions

The expected impact on cost structures of policies such as those on taxation, the environment, incomes and credit is considered in the following review of cost conditions for the period after 1974.

To begin with, taxation policies do not appear to be a significant factor in increased costs and reduced performance, for two reasons. On the one hand, indirect taxes collected when inputs are traded are always reimbursed to the companies, especially when the companies are export-oriented, as were the problem ones. Further, higher indirect taxes apply to imports of final products and not to raw materials [16]. On the other hand, while corporate income taxes were not increased after 1974, marginal increases in tax rates of distributed profits could always be circumvented by reinvesting profits.

Environmental controls also do not appear to have influenced manufacturing business performance. The most serious step toward the control of industrial pollution was taken at a time when the issue of problem firms had been recognized ([17], pp. 23-24). In 1979, the Ministries of Industry and of Labour drafted an Environmental Pollution Control Act and submitted it to Parliament for approval. Among other things, the draft legislation provided for the introduction of stationary source emission standards. Parliament, probably fearing that such controls would be lethal to the already troubled manufacturing sector, did not enact the legislation. The Ministries of Industry and of Public Health did exercise some pressure on firms located in the Athens metropolitan area to curb their pollution, mainly by ordering the burning of lower-sulphur fuels in combustion processes. The absence of cost records on industrial pollution control and of complaints by the business community against environmental regulations also weakens the argument based on the costs of environmental protection.

Unlike taxation and environmental protection measures, however, incomes policies did influence costs, since the Government responded positively to trade union demands for higher wages that had previously been rejected. Therefore wage increases were not a result of increased demand for labour arising from a need to expand the volume of industrial production. The ratio of prices to labour costs decreased significantly between 1974 and 1982 [18].

Credit policies also appear to have played a decisive role in cost determination. Unlike the period before 1974, when credit to manufacturing industry increased in real terms and borrowing costs were lower, credit conditions in the period after 1974 worsened. The real growth rate of credit decreased from 24.5 to 7.1 per cent per annum, while interest rates jumped from 7.5 to 21 per cent. In 1978, especially, interest rates rose to prevent negative effects on the propensity of private depositors to save. The operational structure of the financial system represented an important element of credit policies at the time. The system was essentially controlled by the Government [19]. The processing of applications for new loans was primarily characterized by the inability of the banks to determine final investment costs. That was due to the lack of the following: qualified techno-economic studies submitted by firms to meet application requirements; monitoring of projects to ensure timely execution; and adequate consideration of macro-economic factors such as inflation and interest rates.

Even though both incomes and credit policies did inflate some major cost items, firms could ease the burden by resorting to factor-saving techniques and investment strategies. Cost elasticities with respect to output are presented for problem and non-problem firms in table 2.

Table 2. Cost elasticities in problem and non-problem firms

Major cost item	Problem firms 1973-1976	Problem firms 1977-1980	Non-problem firms 1973-1976	Non-problem firms 1977-1980
Manufacturing	1.28	1.10	1.96	0.76
Labour	1.54	1.49	1.97	1.03
Debt service	1.68	2.07	1.35	0.89
Total	1.36	1.45	1.96	0.83

As reflected in table 2, manufacturing and labour cost elasticities for both groups fell over time, but the debt-service cost elasticity rises only in problem firms. Total cost elasticity in problem firms also appears to increase as a result of debt-service cost effects. Those findings show that both groups were capable of reducing the relative burden of manufacturing and labour cost increases in the period after 1974 through factor substitution. Specifically, entrepreneurs responded to increased costs of raw materials and labour by additional capital investment which changed the factor combination. Unlike their healthy counterparts, however, problem firms financed such projects through fresh credit that was granted almost unconditionally. Interest rates on contracted loans were not fixed, but were allowed to follow the pattern of "market" rates set by the Government. The unexpected rise in interest rates in 1978 was reflected in the higher debt-service cost elasticity of the period from 1977 to 1980, the major impact of which was the reduction in profit margins.

The above findings add fuel to the controversy surrounding the impact of real interest rates on business performance in Greek manufacturing industry. On the one hand, it might plausibly be argued that since real interest rates were either zero or negative because of inflation, business performance should not be associated with them. A counter-argument supported by the present study rejects that claim on two not mutually exclusive counts. First, rising costs leading to a reduction in profit margins have a catalytic effect on a firm's ability to meet increased financial obligations in nominal terms. Secondly, under those conditions, lower net returns, attributed to costs rising faster than prices, when discounted by a higher interest rate, make current project values look less attractive than they were when the project was initially considered.

3. Price conditions

Price formation in Greek manufacturing industry has been affected chiefly by changes in cost components and not by changes in demand ([20], p. 74). Price controls were not introduced until 1979, when problem firms had already been discovered. All price increases were subject to the prior approval of the Ministry of Commerce. Once higher production costs were adequately justified, approval was guaranteed. Between the time of rising costs and that of approved price increases there was, however, a lag that may have had a negative effect on profit margins.

A more essential feature of price conditions during the period after 1974 concerns the open character of the Greek economy. Domestic production is prone to international competition as there are no serious impediments to trade [16]. Therefore, the prices of domestically produced goods move pari passu with those of imported goods. Correlation of import prices with prices of domestically manufactured products is found to be equal to 0.95, which reveals that imports are price leaders. The major conclusion to be drawn is that increased costs experienced by problem firms could not be shifted through higher prices without losing an element of competitiveness and market share.

F. Conclusions

Business failure in Greek manufacturing industry during the late 1970s and early 1980s was a result of both intra-firm and aggregate external causes. Problem firms followed rather risky growth policies, under a management lacking sufficient business acumen, especially weak in the areas of investment and sales strategies. Problem firms desperately needed much higher selling prices to counterbalance the unexpected rise in debt service costs. However, the price leadership exercised by foreign competitors acted as a constraint on price increases. Therefore, the reduction of profit margins in problem firms, and hence their financial disarray, was an inevitable product of their inability to affect prices and control debt service costs. Careful evaluation of government policies during that period appears to show that credit policies played a most crucial role in the determination of costs. Higher interest rates hurt problem firms severely, since their expansionary investment strategies had relied solely on bank credit. As a result, profit margins narrowed to a point where cash flow emerged as the most immediate cause of failure.

The findings presented in this paper complement the limited available literature in various ways. On the one hand, with regard to intra-firm causes, the findings are comparable to those set out by Argenti [1], who attributes failures to bad management. On the other hand, with reference to external influences, unlike Cumming and Saini [2], who found both debt service cost and aggregate demand conditions to be the two principle aggregate external causes of business failure in Japan and the United Kingdom, this paper suggests that it was debt service costs alone that led problem firms in Greek manufacturing industry to their demise. Further, with respect to the Archibald-Baker [4] conclusions, the operational deficiencies of the Greek banking system differ only marginally from those of the non-market allocation activities of federal agencies in the United States. Non-market allocation of federal credit to less efficient firms in that country simply postponed business failures. In Greece, credit policies acted as a catalyst for failure.

The quest for growth in manufacturing industry should not be set within a narrow growth-oriented policy framework. Growth needs to be sustained and protected from the possible negative influences of the firm or of the policy-making machine itself. Policies need to be proactive. The most effective way of preventing business failures in countries such as Greece would be by freeing the banking system so that it could operate under free market conditions, a point repeatedly emphasized in the development literature. In many countries, however, including Greece, such a goal may be essentially unattainable, given their social structures and the powers claimed by central Governments. Under such conditions, the next most suitable policy framework should incorporate failure forecasting and effectively use the results. Moreover, while the introduction of new credit supply rules linked to the type and quality of management seems imperative, government agencies also need to pursue more congruent macro-economic policies.

If the results of the above measures remain inadequate, there exist other ways to cope, at least in part, with the problem of manufacturing business failure. When suspected failures can be attributed to current policy instruments, the best means of avoiding further failure and adverse publicity is to compensate the specific industry for higher production costs. This last proposal raises the following important question for future research: "Did the bail-out programme for failing industries in Greece generate more social costs than compensation initiatives would have generated?".

References

1. J. Argenti, Corporate Collapse (London, McGraw Hill, 1976).

2. C. Cumming and K. Saini, "The macro-economic determinants of corporate bankruptcies in Japan and the U.K.", Research Papers (New York, Federal Reserve Bank, 1981).

3. E. J. Altman, Corporate Financial Distress (New York, John Wiley, 1983).

4. B. R. Archibald and H. S. Baker, "Aggregate business failures and federal credit activity", Public Finance Quarterly, vol. 16, No. 2 (1988).

5. J. Cody and others, Policies for Industrial Progress in Developing Countries (New York, Oxford University Press, 1980).

6. N. Tsoris, The Financing of Greek Manufacture, Studies Series, No. 8 (Athens, Centre of Planning and Economic Research, 1984).

7. T. Giannitsis and C. Vaitsos, Technological Change and Economic Development: Greek Experience and International Prospects (Athens, Gutenberg, 1987) (in Greek).

8. Bank of Greece, "The problem manufacturing firms" (Athens, 1981) (in Greek).

9. K. Nikolaou, "Low profitability and overdebtedness threatens our industry", Economicos Tachydromos, No. 1347, 1981.

10. G. Koutsoumaris, Financing and Growth of Greek Manufacture (Athens, Institute of Industrial and Economic Studies, 1976) (in Greek).

11. J. Papadakis, Loss-making Firms (Athens, Institute for Industrial and Economic Studies, 1983) (in Greek).

12. Centre of Planning and Economic Research, Economic Development Plan of Greece, 1963-1968 (Athens, 1964) (in Greek).

13. B. H. Chenery and I. Adelman, "Foreign aid and economic development: The case of Greece", Review of Economics and Statistics, vol. XLVIII, No. 1 (1966).

14. X. Zolotas, Monetary Equilibrium and Economic Development (Athens, Bank of Greece, 1964).

15. Centre of Planning and Economic Research, Economic Development Plan of Greece, 1968-1972 (Athens, 1968) (in Greek).

16. G. Katsos and N. Spanakis, Industrial Protection and EEC Membership, Studies Series, No. 3 (Athens, Centre of Planning and Economic Research, 1983) (in Greek).

17. J. N. Lekakis, Economics and Air Quality Management in the Greater Athens Area, Studies Series, No. 7 (Athens, Centre of Planning and Economic Research, 1984).

18. Centre of Planning and Economic Research, "Incomes, prices and competitiveness", Report for the Economic Development Plan, 1988-1992 (Athens, forthcoming) (in Greek).

19. A. Papandropoulos, "The market world", Economicos Tachydromos, No. 1618, 1985 (in Greek).

20. V. Economopoulou, Price Formation in Greek Manufacture, Studies Series, No. 14 (Athens, Centre of Planning and Economic Research, 1984) (in Greek).

 18256

THE POTENTIAL FOR TRADE IN GOODS AND SERVICES BETWEEN DEVELOPING COUNTRIES: AN EXAMPLE FROM THE PETROCHEMICALS INDUSTRY

Elizabeth Parsan*

Introduction

The need to expand trade between developing countries, or what is popularly termed South-South trade, has gained increasing prominence in recent years. Several factors, both external and internal to the South, have contributed to that trend. On the one hand, it has been observed that in the light of changes taking place internationally, North-South trade cannot be relied upon in the future as an "engine of growth" for developing countries [1]. On the other hand, current developments within the South itself now create dynamic opportunitites for trade within the group.

Two external trends have been the increasing tendency towards protectionism in the North and the slow-down in the growth of the economies of developed countries. Besides the traditional tariff and non-tariff barriers, developed countries have introduced a variety of new trade restrictions such as voluntary export restraints and orderly marketing arrangements. Such measures, together with the reduced rates of growth in developed countries, have adversely affected the growth of South-North trade.**

The more powerful arguments for South-South co-operation, however, are based on the emerging and dynamic complementarities in the South. Over the past three decades, production and trading structures in the South have become more diversified, and there now exists expanded potential for trade in both goods and services. In addition, the competitiveness of many countries of the South has increased in a number of areas, as evidenced by their ability to provide a strong challenge to their competitors in the North. Therefore, structural changes taking place both internationally and within the South now render South-South co-operation imperative and beneficial.

In this paper an attempt is made to explore some of the emerging opportunities for South-South trade in an area which has been identified as having great potential - the petrochemicals industry ([3], [4]). Those opportunities are examined within the context of trade between two specific countries - Brazil and Trinidad and Tobago. Their case is interesting from both an industry and a country point of view.

*University of the West Indies, Saint Augustine, Trinidad and Tobago.

**Whereas growth rates in developed countries had been sustained at relatively high levels up to 1974, growth in the more recent period has not averaged more than 4 per cent per year [2].

The petrochemicals industry is one in which many oil-rich, developing countries have expanded their productive structures since the mid-1970s, and in which they have a strong comparative advantage. It is also one in which South-North trade has been adversely affected because of growing protectionism in the North [5]. The countries being considered are at different levels of development. Brazil belongs to the group of newly industrializing countries (NICs),* while Trinidad and Tobago is one of the less industrialized countries of the South. However, as will be seen later, they exemplify the possibilities for trade, in both goods and services, between those two groups within the South.

In section A, some of the theoretical considerations underlying the empirical study are first discussed. In section B, the structural features of the petrochemical industries in both Brazil and Trinidad and Tobago are examined, and potential areas for trade identified. A detailed review of the scope for trade in those areas follows in section C, and the benefits likely to accrue to both trading partners are considered in section D. Finally, the main conclusions of the paper are drawn, and some policy recommendations made, in section E.

A. Some theoretical issues

Although there exists no formal theory of South-South trade, Stewart [6] has noted that many of the existing trade theories which emerged to explain North-North or South-North trade in fact apply to South-South trade. The extended Heckscher-Ohlin theory and the technology theories, which provided a guide for empirical study, are dealt with in this section.

1. Extended Heckscher-Ohlin theory

The simple Heckscher-Ohlin theory, based upon two countries, two factors and two goods, holds that inter-country differences in factor endowments provide a basis for trade. Therefore, countries with a relative abundance of capital, such as those of the North, will tend to export capital-intensive goods to countries with a relative scarcity of capital, such as those of the South.

According to this simple theory then, trade is likely to flow from North to South and vice versa [7].

An extended version of the Heckscher-Ohlin theory, however, incorporates multilateral trade flows and provides insight into the potential for South-South trade ([8], [9]). It states that a country will have a comparative advantage in a range of goods on the basis of its relative endowment. But to the extent that these goods may be produced with different factor intensities, the comparative

---

*The term "NICs" is used extensively to describe developing economies, be they countries, provinces or areas, where there has been particularly rapid industrial growth.

advantage of a country will vary with the difference between its own factor endowments and those of its trading partners.

The predictions of the extended theory are important, for they imply that a country will trade in both North-South and South-South directions. For example, if a model with both physical and human capital is considered, economies such as the NICs will export to the South more skill- and capital-intensive goods than they will to the North. At the same time, they will also import from the North more skill- and capital-intensive goods than they will import from the South.

Hence, to the extent that the factor content of goods traded within the South differs from that of goods exported to the North, special gains may be expected from South-South trade. That is, since the latter is associated with more skill- and capital-intensive trade, South-South trade provides for the accumulation of both physical and human capital by countries of the South. Those resources, in which the South is particularly deficient, therefore have great significance within a developmental perspective.

In addition, there are important interactive effects. As dynamic exporters in the South begin to reap the benefits from South-South trade, the growth of their economies will be stimulated. These growing economies will in turn become important markets for other countries of the South, and will provide a further stimulus to economic growth in the latter.

2. Technology theories

The technology theories, which include the technology-gap and the product-cycle theories, also provide important insights into the potential benefits of South-South trade. These theories emphasize technological innovation as an important factor in influencing trading patterns. In the technology-gap model developed by Posner [10], leads and lags in technological innovation are seen as determining trade patterns, whereas the product-cycle model [11] attempts to explain trade patterns on the basis of stages of a product life.

The product-cycle theory, which is the more widely used, proceeds from the premise that a product typically passes through three stages - early, growth and mature. During the first phase, new products that tend to be highly skill-intensive are developed, and countries with abundant supplies of skilled labour are most likely to have an advantage in the production and export of those products. In the second phase, as demand grows and know-how spreads, countries with less technical expertise begin to produce and market the products. Finally, in the mature phase, the product becomes completely standardized, and the production technology becomes internationally diffused.

The technology theories emerged primarily to explain North-North trade. In the framework offered, developing countries are viewed essentially as receivers of technological change. They import "new-technology" products from the North which normally command high prices, and engage in production and export only later when the technologies become mature and standardized. These theories therefore rule out South-South trade.

However, if countries of the South are viewed as innovators themselves, then the technology theories can be made to apply to South-South trade. In such a case, they would be able to reap the benefits of technical change in much the same way as the North has been able to. Indeed, existing evidence suggests that developing countries such as Brazil, India and the Republic of Korea are becoming important innovators and initiators of technological change [12].

The innovators could profit from the higher prices normally paid for new products. More importantly though, gains could be made from the provision of technologies that may be more suited to the factor characteristics and income levels of the South. Indeed, in many cases, technologies imported from the North are highly capital-intensive, and do not suit the labour-surplus economies of the South. Further, new products from the North tend to reflect increasingly high-income characteristics, and may be inappropriate for countries of the South, given their lower income levels.

B. Potential areas for trade between Trinidad and Tobago and Brazil in petrochemicals

1. The basis for trade

The features of the economies of Brazil and Trinidad and Tobago are different, and the two countries are at varying levels of development. Yet it is precisely those dissimilarities that create a basis for trade. The Brazilian economy is large and highly diversified. The country has an estimated population of 135.6 million and a land area of 8.5 million square kilometres, and, with a per capita income of $1,640, is considered an "upper-middle-income" country [13]. Further, its natural resource base is large, and its production and exports cover a wide range of both primary and manufactured goods as well as services.

There is one area in which the Brazilian natural resource base is deficient, and that is oil. In 1986, 25.5 per cent of Brazilian imports consisted of oil, and 7.1 per cent of energy-intensive products such as petrochemicals [14]. Oil imports have actually been declining since 1985, following oil and natural gas discoveries in the country. However, petrochemical imports continue to grow,* because oil and natural gas, two important base materials for the petrochemical industry, have many alternative uses in Brazil.

The Trinidad and Tobago economy, on the other hand, exhibits different features from that of Brazil. Although its per capita income is relatively high ($5,120 in 1986), it is a small country

*Between 1985 and 1987, imports of petrochemicals grew on average by 25.2 per cent annually [14].

with a population of 1.2 million and a land area of 5,128 square kilometres [13]. Further, whereas its resource base is narrow, it is abundantly endowed with large quantities of oil and natural gas, on which its production and trade is based. In 1987, 73 per cent of the total exports of Trinidad and Tobago consisted of oil, and 14 per cent of petrochemicals [15].

It therefore appears that differences in endowments of oil and natural gas resources could provide a basis for trade between the two countries. Since imports of energy-intensive products such as petrochemicals are large in Brazil, and Trinidad and Tobago exports substantial amounts of those products, there is scope for trade in the petrochemicals industry.

Another basis for trade can be found in the fact that Brazil has been able to upgrade its technical skills, knowledge and experience over the past few decades, and is fast emerging as an important exporter of technology. Brazilian technology exports cover the supply of technical services in a variety of areas, and are related both to the sale of capital goods abroad and to the delivery of construction projects. Since the country is particularly strong in the oil industry [16], trade in technology is perhaps another area be explored.

2. Structural features of the petrochemical industries in Brazil and Trinidad and Tobago

Among developing countries Brazil has the largest and most highly diversified petrochemical industry. Table 1 shows that it produces all the major basic and final petrochemical products on a very large scale. Its production of ethylene, which is the dominant product in the industry, and which provides the base for plastics such as polyethylene and polyvinyl chloride, exceeds 1 million tonnes annually.

By contrast, the industry of Trinidad and Tobago produces a narrow range of products. Its output consists mainly of ammonia, methanol and urea, which are produced in large quantities. The industry does not produce ethylene or any of the plastics, even though the feedstocks, both oil and natural gas, are widely available domestically.* However, it does have a small downstream plastics processing industry which utilizes imported plastic resins.

The major part of petrochemical production in Brazil is based on oil refinery fractions, feedstocks which are known to be less economic than natural gas. Further, some of the technologies used are of an older vintage. For example, roughly one half of ammonia production is based on steam-reforming of naphtha and partial oxidation of fuel oil, which are high-cost products and processes [18].

*Although proven reserves of oil and natural gas in Trinidad and Tobago represent, respectively, 0.06 per cent and 0.3 per cent of total proven world reserves, its production-reserves ratios are low, and local consumption is small (see [17]).

Therefore, despite large feedstock subsidies, production costs are high in Brazil as compared with oil-rich countries such as Trinidad and Tobago.

Table 1. Petrochemical product structure, Brazil and Trinidad and Tobago, 1984

Product	Brazil		Trinidad and Tobago	
	Capacity (thousand of tonnes per year)	Percentage of total	Capacity (thousand of tonnes per year)	Percentage of total
Basics				
Olefins		—	—	—
Ethylene	1 263	27.9	—	—
Propylene	697	15.4	—	—
Butadiene	258	5.7	—	—
Aromatics				
Benzene	420	9.3	30	1.7
Toluene	114 a/	2.5	49	2.7
Xylene	354 a/	7.8	35	1.9
Ammonia	1 2440	27.4	1 300	71.7
Methanol	180	4.0	400	22.0
Total basics	4 526	100.0	1 814	100.0
End products b/				
Plastics	1 408	75.3	--	--
Styrene-butadiene rubber	188	10.1	--	--
Synthetic fibres	274	14.6	--	--
Total	1 970	100.0		
Fertilizers c/	1 747	—	335	—

Sources: World Petrochemicals (Stanford, Stanford Research Institute International, 1985); Ministry of Energy and Natural Resources, Annual Energy Report 1985 (Port of Spain, Control Statistical Office Printing Unit, 1986); Petrofertil, Annual Report 1985 (Rio de Janeiro, 1986).

a/ Figures include only productive capacity at the three petrochemical complexes.

b/ Figures are for production.

c/ Figures include only nitrogen fertilizers.

The latter is reflected by the fact that production costs of ammonia in Trinidad and Tobago are 25 per cent lower than those in Brazil. The disparity in methanol production costs is far greater, as illustrated by the fact that the price of methanol in Brazil is more than three times the production cost of that of Trinidad and Tobago [18]. The use of modern, large-scale technologies that permit economies of scale, and the fact that the industry is based almost exclusively on low-cost natural gas, contribute towards the lower production costs in Trinidad and Tobago.

Production of petrochemicals in Brazil is geared principally to the domestic market. Exports are limited to a few products, and only plastics such as polyethylene are exported in large quantities. Imports, however, have been large, especially products such as ammonia and urea, in which Trinidad and Tobago has a stong comparative advantage. For example, in 1980, a year in which imports of those products peaked, ammonia imports amounted to 235,000 tonnes, or 35 per cent of consumption, while corresponding figures for imports of urea were 579,000 tonnes, or 68 per cent of consumption [14]. Petrochemical production in Trinidad and Tobago, however, is geared almost entirely to the export market, about 90 per cent of its output being exported.*

The direction of trade in both Brazil and Trinidad and Tobago is significant. Despite the wide availability of low-cost products in Trinidad and Tobago, Brazilian imports of petrochemicals have been largely from a higher cost supplier in the North, the United States of America.** Similarly, in spite of the close proximity of Brazilian markets, exports of petrochemicals from Trinidad and Tobago have been to the North, mainly to the United States and Western Europe. Trade between Brazil and Trinidad and Tobago in petrochemicals has been almost non-existent in the past. Only recently, during the 1980s, have small amounts of ammonia, urea and methanol been exported to Brazil from Trinidad and Tobago.*** The reasons for this pattern of trade will be discussed later.

Two other important features should be noted. First, whereas the bulk of petrochemical production in Brazil is State-owned, in the case of Trinidad and Tobago, transnational corporations based in the North have dominated the petrochemical industry. Their dominance has largely contributed to the skewed pattern of production in the industry, because they have primarily been interested in utilizing the country's low-cost natural gas, the technical properties of which make it most suitable for manufacturing products such as ammonia, urea and methanol. Little attention has been given to the use of oil refinery fractions or other feedstocks such as natural gas liquids, which could have allowed a wider range of production, including, for example, ethylene and its derivatives.

---

*In fact, Trinidad and Tobago is the world's second largest exporter of ammonia [19].

**United States production costs of ammonia, for example, are 27 per cent higher than those in Trinidad and Tobago (see [18]).

***In 1987, Trinidad and Tobago exported 35,600 tonnes of ammonia and 35,300 tonnes of urea to Brazil [15].

The second point relates to the accumulation of technological expertise. Whereas the Trinidad and Tobago industry relies almost exclusively on foreign technological expertise, over the past three decades Brazil has managed to develop a substantial technological capability in petrochemicals manufacture, acquired from the experience of constructing three large-scale petrochemical complexes and a number of individual plants.

Table 2 shows how Brazil has progressed in providing technological inputs to the local industry. During the construction of the Sao Paulo Complex between 1965 and 1972, only 50 per cent of the technological inputs required were provided locally. However, that proportion rose to 75 per cent between 1978 and 1982 during the construction of the Southern Complex, with all preliminary studies, detailed engineering and construction and assembly work undertaken locally. There are some noticeable gaps in Brazilian technological expertise, in the areas of process licensing, basic engineering and, to a lesser extent, technical co-operation capabilities. Nevertheless, the Brazilian industry considers itself to be in a good position to export petrochemical technologies, and this area has been earmarked for development with the South.

3. Potential areas for trade

Given the structural features of the Brazilian and the Trinidad and Tobago petrochemical industries described above, two sets of trading possibilities appear to have potential. First of all, the substantial export capability and strong comparative advantage enjoyed by Trinidad and Tobago in the production of ammonia, urea and methanol, coupled with the fact that Brazil imports those products in fairly large quantities, imply that there may be scope for profitable trade in those products. One trading possiblity therefore involves the export of selected petrochemical products - ammonia, urea and methanol - from Trinidad and Tobago to Brazil.

Secondly, given the wide availability of both oil and natural gas in Trinidad and Tobago, and given the fact that its petrochemical base is narrow and does not include any production, such as that of plastics, in the olefins stream, and that Brazil is now in a position to export petrochemical technologies, a second possibility can be explored, namely the export of petrochemical technologies by Brazil to Trinidad and Tobago to further develop a plastics industry in the latter country.

The establishment of a plastics industry in Trinidad and Tobago could involve the production of ethylene, which would be further processed to produce various plastic resins such as polyethylene and polyvinyl chloride. Such an industry could be geared primarily to export, but could also feed into the small local plastics processing industry. It could also draw upon both oil refinery fractions and natural gas.

Table 2. The Brazilian petrochemical industry: local supply of technological inputs
(Percentages)

Petro-chemical complex	Category of inputs							
	Preliminary studies	Process licensing	Basic engineering	Detailed engineering	Technical co-operation	Construction and assembly	Capital goods	Proportion of local content
Sao Paulo (1965-1972)	50	-	-	10	5	100	45	50
North-east (1972-1978)	100	-	-	70	30	100	60	65
Southern (1978-1982)	100	-	-	100	30	100	75	75

Source: [27].

C. Scope for trade in petrochemicals between Brazil and Trinidad and Tobago

1. Trade in goods: exports of ammonia, methanol and urea from Trinidad and Tobago to Brazil

The development of exports from Trinidad and Tobago to Brazil in ammonia, methanol and urea products will depend largely on the following two sets of factors: future Brazilian demand for imported petrochemicals; and the competitiveness of Trinidad and Tobago products in Brazil.

(a) Brazilian demand for imports

The future demand for petrochemicals in Brazil will depend to a large extent on growth of the economy. In the past, when economic growth was rapid, growth of consumption, and hence of imports of ammonia, urea and methanol, was high. From 1974 to 1980 the economy grew on average by 7 per cent per year in real terms; the demand for ammonia, urea and methanol grew on average by 25 per cent, 43 per cent and 28 per cent annually; and imports accounted for about 40 per cent of consumption. Annual import levels increased to as high as 235,000 tonnes for ammonia and 430,000 tonnes for urea, but methanol imports did not exceed 13,000 tonnes ([19], [20]).

From 1982 to 1985, however, there were drastic declines in petrochemical consumption and imports, as growth in the economy slowed down. During that period, annual growth of the economy in real terms averaged 1.8 per cent, and annual growth in the consumption of ammonia, urea and methanol dropped to an average of, respectively, 14 per cent, 5 per cent and 4 per cent. Annual imports fell drastically, to as low as 16,000 tonnes of ammonia and 3,000 tonnes of urea. Increased domestic availability of petrochemicals contributed to the fall in imports ([19], [20]).

Since 1985, however, economic growth has resumed in Brazil, and higher annual average growth rates of 6.4 per cent were recorded from 1985 to 1987 [2]. At the same time there has been an increase in the growth of consumption, and hence of imports of ammonia, urea and methanol. For example, from 1985 to 1986 imports of those products increased from 17,000 tonnes, 16,000 tonnes and 21,000 tonnes, to 51,000 tonnes, 165,000 tonnes and 42,000 tonnes, respectively [14]. If economic growth continues to be rapid in the future, then imports of petrochemical products are likely to be large.

Besides higher economic growth, other factors may influence increased consumption levels and imports in the future. Fertilizer prices are now lower, compared to previous years, and there is increased availability of farm credit. Further, major agricultural programmes have been initiated. Those factors, together with the fact that fertilizer use is currently very low in Brazil, point to potentially large increases in consumption of ammonia and urea in the future [18]. Growth in methanol use may also increase considerably, depending upon future government plans to export methanol-based products.

On the basis of past trends, and taking into account the above-mentioned factors, it is estimated that growth in consumption of ammonia and urea in the medium term (up to the year 2000) is expected to be around 7 per cent and 8 per cent per annum, respectively. Annual growth in the consumption of methanol is projected at a lower rate of 6 per cent.

In order to estimate the potential for imports, however, the expansion of domestic supplies in Brazil must be considered. There have been large increases in petrochemical production over the past decade, and plans are being made to further expand capacity. Two new plants for annual production of 271,000 tonnes of ammonia and 167,000 tonnes of urea are expected to be constructed in the future. Taking into account those production plans and projected consumption, demand can be expected to outstrip supply in Brazil in the future. By the year 2000, annual imports of ammonia, urea and methanol are likely to reach 946,000 tonnes, 1.1 million tonnes and 100,000 tonnes, respectively.

(b) Competitiveness of Trinidad and Tobago petrochemical products in the Brazilian market

Trinidad and Tobago, since its petrochemical production costs are lower than in Brazil, should be able to compete successfully against Brazilian domestic production. However, other costs, such as transport and tariffs, must be considered. As reflected in table 3, shipping costs form a small proportion of the landed price of Trinidad and Tobago products. However, tariff rates have been high in Brazil, and this has severely curtailed the ability of Trinidad and Tobago to compete in that country's market. The tariffs on ammonia, urea and methanol imports have been 45 per cent, 15 per cent and 45 per cent, respectively.

A comparison of Trinidad and Tobago's landed prices with that of Brazilian market prices reveals that the former were 22 per cent higher for both ammonia and urea. Only in the case of methanol was it able to compete effectively. It must be noted that besides tariffs, other forms of protection are given to the Brazilian industry, through subsidies on inputs such as natural gas, price controls and a discretionary import licensing programme.

Calculation of effective protection rates, which incorporate subsidies on inputs, revealed much higher levels of protection, up to 267 per cent for ammonia and 169 per cent for methanol. Tariff equivalents of price controls were also estimated at 72.3 per cent, 44.1 per cent and 0.7 per cent for ammonia, urea and methanol, respectively [18].

Besides tariff and non-tariff barriers, an inadequate trading infrastructure, particularly in marketing and export-credit facilities, have adversely affected Trinidad and Tobago competitiveness in the past. In fact, it is largely because of those factors that higher-cost producers in the United States have been able to capture the Brazilian market.* Table 3 shows that the landed prices of Trinidad and Tobago petrochemicals in Brazil are far lower than those of the United States.

Recent developments suggest, however, that the competitiveness of Trinidad and Tobago petrochemicals is likely to improve considerably in the future. First of all, marketing efforts in Trinidad and Tobago have become more aggressive, as the traditional markets in the North have begun to shrink with increasing protectionism and lower growth. For example, in July 1986 Trinidad and Tobago was accused of dumping urea on the European market. Anti-dumping duties were subsequently levied on its imports [22].

Secondly, domestic availability of export-credit facilities has increased, following the establishment of the Trinidad and Tobago Export Credit Insurance Company. In addition, trade finance is now available through the Latin American Export Bank. Finally, and most importantly, Brazil has begun to dismantle some of its trading barriers. In early 1989, tariff rates for imports of ammonia were reduced to 10 per cent [23]. At this rate, the landed price of Trinidad and Tobago ammonia is roughly equivalent to the market price in Brazil. If such a relaxation of trade barriers continues in the future, and the demand for imports of petrochemicals by Brazil conforms to projections, then there may be considerable scope for trade between Brazil and Trinidad and Tobago.

2. Trade in services: exports of petrochemical technology from Brazil to Trinidad and Tobago

In investigating the possiblities for the export of petrochemical technologies from Brazil to Trinidad and Tobago, consideration must be given to the extent to which Brazil can provide technologies required by the Trinidad and Tobago industry, and its ability to compete with suppliers from the North.

As mentioned earlier, Brazil has had wide experience in building, operating and managing petrochemical plants, and considers itself to be in a good position to export technology. However, despite Brazil's strong technological capabilities, there are a few areas in which it is seriously deficient. These include the critical areas of process licensing and basic engineering - areas which involve the "know-how" or "core" technologies - and, to a lesser extent, technical co-operation. The question then arises as to how serious those deficiencies are in limiting the role of Brazil as a technology supplier.

The above deficiencies are not likely to pose a serious problem, given the nature of, and the market for, such technology. It is generally recognized that the technology required for most basic petrochemicals such as ethynlene is mature, standardized and widely available, so that there exists a plurality of technology

*See Parsan [21] for an analysis of the influence of these factors.

Table 3. Competitivity of selected petrochemical products from Trinidad and Tobago and the United States in the Brazilian market, 1984
(Dollars per tonne)

Product	Production cost +5 per cent a/	Shipping cost	c.i.f. price	Tariffs b/	Landed price	Market price
		Trinidad and Tobago				Brazil
Ammonia	96.60	20.00	116.60	52.47	169.07	132
Urea	140.00 c/	17.00	157.00	23.55	180.55	140
Methanol	103.95	15.00	118.95	53.52	172.47	350 c/
		United States				
Ammonia	157.50	24.00	181.50	81.68	263.18	132
Urea	170.00 c/	27.55	197.55	29.63	227.18	140
Methanol	222.50	20.00	242.50	109.13	351.63	350 c/

Sources: [21], for production cost data; industry estimates and [14], for shipping costs; Petrofertil (see [14]), for Brazilian market prices of ammonia and urea; V. Menezes Pereira, Oportunidades de Produção de Metanol no Brasil (Rio de Janeiro, Petroquisa, 1984), for Brazilian methanol prices; and [23], for tariff rates.

a/ +25 per cent for the United States.

b/ Tariff rates for ammonia and methanol are 45 per cent, and for urea 15 per cent of c.i.f. value.

c/ f.o.b. prices.

sources [24]. This is also the case for end-products such as polythylene and polyvinyl chloride. Competitive technologies now exist for high-density polyethylene and can be easily obtained from several companies [25].

The highly competitive market for petrochemical technologies therefore suggests that it may be preferable to source the most suitable technology at the cheapest possible price, rather than to own the technology. The question of sourcing is particularly important in the present case, because what the Trinidad and Tobago industry needs is not a specific technology, but an appropriate mix of technologies that will enable it to utilize both its oil and natural gas ref . ·s. The ability to select and evaluate suitable technologies i critical.

Given Brazi 's experience in building and operating plants based on both oil refinery fractions and natural gas, together with its expertise in sourcing technologies, it is likely to be superior to a technology supplier from the North. Brazilian production of ethylene is based on many different raw materials and processes, which include cracking of naphtha, gas oil and natural gas, as well as dehydration of sugar alcohol. It is therefore likely to be well-equipped to provide technological expertise in a variety of areas.

The skilful way in which Brazil has sourced petrochemical technologies from countries of the North and managed to assimilate such technologies has been described by Sercovich [26]. Its success has involved extensive and rigorous searches for the best package and a careful "de-packaging" of technology. Its growing ability to source technologies is related to an important issue involving the costs of acquiring technology.

Teixeira [27] has noted that as Brazil pursued its aggressive policy to acquire technological expertise, and learning progressed in the industry, it was able to acquire technology on more favourable terms. That was confirmed by the Head of the Technical Division at Petroquisa (the State-owned company responsible for the petrochemical industry in Brazil), when he commented that:

> "A 'high' price had to be paid in the installations of the first petrochemical plants ... Only experience allowed us to improve upon subsequent contracts. So much so that various firms in operation contracted technology from abroad with the right to free use" ([28], p. 39).

Other indirect evidence suggests that not only is technology provided by Brazil likely to be low-cost, but also Brazil may be more willing to transfer technology compared to a supplier of the North. The following was noted in connection with the establishment of a plant for brick- and block-making in Nigeria, for which Brazil provided technology:

> "Technology was freely imported, layout plans were also provided without fee, and no charge was made for Brazilian engineers' travel to commission and start up

plants, provided accommodation was provided and a minimum number of machines were involved. Supply of equipment included ' years' spare parts for fast-wearing items, but any .irs to the machines involving visits by Brazilian nvolved the Nigerian owner in meeting all expenses" ([29] p. 5).

Despite its growing technological capabilities, however, Brazil has not been very successful in exporting its technologies, given strong competition from suppliers of the North. The experience of Brazil in exporting petrochemical technologies has been limited to the conduct of pre-investment studies and the supply of technical advice for the establishment of petrochemical plants, primarily to countries in Latin America. However, as shown by Sercovich [26], a country may opt for a technology supplier that is less well-known and has less experience in exporting, if it feels that such a supplier may be more willing to transfer technology. Indeed, Brazil made that choice in its search for a supplier of plastics technologies for its industry.

Finally, the extensive structures for research and development that are required to foster major technological change are only now being built up in Brazil [28]. However, given the similar conditions of Brazil and Trinidad and Tobago, the latter country may be able to benefit from technical changes in Brazil, including measures to reduce costs and increase production, and better use of inputs as a result of improved operating conditions and equipment modifications.

More importantly, as noted by Sercovich [26], because of the efforts currently being taken in Brazil to develop its technological capabilities, it is one of the few developing countries that may be able to reach the forefront of technology in the future. Given its growing capacity to source and provide technologies that may be especially useful to the industry of Trinidad and Tobago, the scope for trade in this area appears to be great.

D. Potential benefits from trade

The above discussion suggests not only that there is potential for trade in goods and services between Brazil and Trinidad and Tobago, but also that there are substantial benefits to be made. Those benefits, together with the potential losses resulting from the development of South-South trade, are considered in this section. They are viewed within the context of alternatives available through other trading links, namely South-North trade.

From the point of view of Trinidad and Tobago, important benefits can be made from the export of ammonia, urea and methanol to Brazil, but those benefits are mainly static. They include possible benefits from expanded production and market diversification. The latter is very important, given the growing levels of protectionism in the North and the need to seek wider, high-growth markets. Indeed, in recent years, the growth in demand for petrochemicals in developed countries has been slower than that in developing countries, and projections suggest a continuation of that trend in the future [25].

Brazil can reap similar static benefits by importing petrochemical products from Trinidad and Tobago. Since product from the latter country are more cost-competitive than those from the United States, Brazil could acquire lower-cost imports and considerably increase its consumption.

Indeed, a reorientation away from North-South trade would not lead to costly trade diversion for Brazil, but would encourage fruitful trade expansion.

In the case of both Brazil and Trinidad and Tobago, the benefits would have to be weighed against losses likely to arise from the lack of finance, since both countries have a huge debt burden.* In recent years, Trinidad and Tobago exports to Brazil, in particular, its exports of iron and steel products during the early 1980s, have been hampered by payment problems on the part of Brazil. If Brazil is unable to pay for its imports, then losses may result. Similarly, if Trinidad and Tobago is not in a position to provide export credit to Brazil, then Brazil may find it more advantageous to continue importing from the United States.

The export of technology from Brazil to Trinidad and Tobago for the development of a plastics industry could also bring important, more dynamic benefits to both countries. For Trinidad and Tobago there are general benefits associated with the establishment of a plastics industry. There are also specific benefits to be gained from importing technology from a supplier of the South. Each set of benefits will be discussed in turn.

Since a plastics industry would encourage the use of hitherto untapped raw materials in Trinidad and Tobago, benefits would result from the utilization of natural resources. Moreover, since plants will be geared for export, important foreign exchange earnings may be generated. In addition to those benefits, other indirect benefits could accrue through the creation of inter-industry linkages. The plastics industry links backwards to the oil industry, and forwards to plastics processing industries.

It has been suggested that domestically produced plastics resins in Trinidad and Tobago are likely to be cheaper than those currently imported [18]. Domestic availability of plastics resins could create an incentive to scale up the levels of production in the small, local downstream plastics processing industry. Benefits could therefore be gained through economies of scale.

---

*In 1986 Brazil recorded a total external debt of $110.7 billion, the servicing of which accounted for 33.2 per cent of its exports of goods. Trinidad and Tobago has a smaller debt ($1.4 billion) and lower debt-service ratio (15.8 per cent), but given its small size, its debt situation is perhaps equally burdensome [30].

The provision of technology by Brazil can bring other benefits. Brazil is likely to be favourable to the use of both natural gas and oil refinery fractions for plastics production in Trinidad and Tobago, unlike a supplier from the North who may be interested only in the utilization of natural gas. The possiblity and terms of transfer of technology are also likely to be more favourable than in the case of a supplier from the North. Further, given the similar conditions of the two countries, benefits could be gained through the provision of technology that may be appropriate to the needs of Trinidad and Tobago. Those benefits are likely to outweigh losses that might arise from Brazil's lack of experience in exporting such technologies.

Brazil could also benefit greatly from the export of its technologies through the process of technological learning and the upgrading of its skills. By providing a larger market for products and processes that may have been specially adapted for its own conditions, the export of technologies could also stimulate technological change in Brazil.

In view of all the above-mentioned factors, potential benefits are likely to far outweigh potential losses for both trading partners.

E. Conclusions and policy recommendations

The purpose of this paper was to explore the potential for South-South trade by reviewing the possibilities for trade between Brazil and Trinidad and Tobago in the petrochemicals industry. This case of South-South trade is a specific one, hence no generalizations can be made. However, it does shed light on some of the wider issues involved in a strategy of South-South trade.

It has been shown how countries of the South which are at different levels of development could engage in mutually beneficial trade, both in goods and services. Existing complementaries in the petrochemical industries of Brazil and Trinidad and Tobago produce the potential for trade in specific areas. Largely because of its natural resource endowments, Trinidad and Tobago has a cost advantage in the production of certain petrochemicals, in particular ammonia, urea and methanol, which Brazil imports in sizeable quantitites. On the basis of the extended Heckscher-Ohlin theory, a case has been made for the export of those petrochemicals from Trinidad and Tobago to Brazil.

Trade in services has also been considered. It has been pointed out that Brazil has acquired considerable technological expertise in petrochemicals manufacture, and is therefore in a position to assist Trinidad and Tobago in expanding its petrochemical industry. The area selected for development is the plastics industry. Product-cycle considerations have guided the identification of a second area for trade, the export of technology by Brazil to Trinidad and Tobago.

The benefits from trade involving the export of petrochemicals are mainly static. That is, consumption benefits could be reaped by Brazil through access to lower-cost products, whereas benefits to Trinidad and Tobago could be in production and the diversification of trade. In the light of growing protectionism in the North, the latter benefits assume special significance.

The benefits from trade in technology are more dynamic. For Brazil they include benefits through technological learning, the upgrading of skills and the stimulation of technical change. For Trinidad and Tobago, general benefits are expected from the addition of new production facilities, but specific benefits are more important. The latter include wider resource utilization, a greater transfer of technology, especially more appropriate technology, and more favourable terms of transfer. Benefits from South-South trade are likely to be far greater than those arising from trade between the North and the South.

While the scope for trade in the petrochemical industry is large, there remain important constraints to such trade. Future demand is likely to outstrip the supply of specific petrochemicals in Brazil. Hence future imports should be large, and the scope for trade great. However, the existence of tariff and non-tariff barriers are a major obstacle to trade, notwithstanding the fact that there have been some recent attempts to lift those barriers.

High Brazilian tariff rates for petrochemicals render most of Trinidad and Tobago products uncompetitive in the Brazilian market. In addition, non-tariff barriers such as the discretionary licensing programme are prevalent, as well as various other kinds of protection, including input subsidies. Institutional constraints are harder to analyse, but indirect evidence suggests that in the past the lack of export credit facilities and the dominance of transnational corporations have posed important barriers to the development of trade between Brazil and Trinidad and Tobago. Similarly, there is great scope for trade in technology, with one important constraint. Brazil has had little experience in exporting petrochemical technologies, and is unknown as a provider of such technologies. It may therefore be at a severe disadvantage when competing against suppliers from the North.

In light of the above, one clear policy recommendation appears to be that trade barriers should be dismantled. Indeed, if tariffs are lowered and non-tariff barriers removed, there will be benefits for both trading partners. Moreover, given the fact that petrochemicals imported from the North into Brazil are higher-priced than those from Trinidad and Tobago, fruitful trade expansion rather than trade diversion can be expected to take place between the two countries.

A second policy recommendation is that both countries should strengthen their trading infrastructure by augmenting the supply of resources for export credit and implementing more aggressive marketing systems. Accordingly, counter-trade possibilities could be explored in order to alleviate the problem of insufficient trade finance. Information systems that can inform prospective buyers and sellers of available opportunities also need to be developed.

A third policy recommendation is that both countries should explore possibilities for joint-venture arrangements in the petrochemicals industry. One such arrangement could draw upon the natural resources of Trinidad and Tobago and the technological expertise of Brazil. Indeed, South-South joint ventures are considered to be particularly important, since they are known to accord special benefits in line with the development needs of the parties involved.

References

1. Arthur Lewis, "The slowing down of the engine of growth", American Economic Review, vol. 70, No. 4 (September 1980), pp. 555-564.

2. World Economic Survey 1988 (United Nations publication, Sales No. E.88.II.C.1).

3. "Opportunities for cooperation among the developing countries for the establishment of the petrochemical industry", Sectoral Working Paper, Series No. 20 (UNIDO/IS.376).

4. Thomas R. Stauffer, "Energy-intensive industrialization in the Middle East", Industry and Development (United Nations publication, Sales No. E.85.II.B.3), No. 14, pp. 1-35.

5. "Tariff and non-tariff measures in the world trade of petrochemical products" (UNIDO/15.573).

6. Frances Stewart, "Recent theories of international trade: some implications for the South", in Monopolistic Competition and International Trade, H. Kierskowski, ed. (Oxford, Oxford University Press, 1984).

7. Frances Stewart, "The direction of international trade: gains and losses for the Third World", in A World Divided: The Less Developed Countries in the International Economy, Gerald Helleiner, ed. (Cambridge, Cambridge University Press, 1976).

8. Anne Krueger, Growth, Distortions and Patterns of Trade Among Many Countries, Princeton Studies in International Finance Series No. 40 (Princeton, Princeton University Press, 1976).

9. Alan Deardorff, "The direction of developing country trade: examples of pure theory", paper presented at World Bank Conference, Université Libre de Brussels, Brussels, 28 February - 1 March 1983.

10. Michael Posner, "International trade and technical change", Oxford Economic Papers, vol. 13, 1961, pp. 757-778.

11. Raymond Vernon, "International investment and international trade in the product cycle", Quarterly Journal of Economics, vol. 80 (1966), pp. 190-207.

12. Sanjaya Lall, Developing Countries As Exporters of Technology (London, Macmillan, 1982).

13. International Bank for Reconstruction and Development, World Development Report 1987 (Washington, D.C., World Bank, 1988).

14. Banco do Brasil, Comércio Exterior (Rio de Janeiro) 1981-1988.

15. Overseas Trade (Port-of-Spain, Central Statistical Office, Printing Unit, 1987).

16. Francisco Sercovich, "Exports of technology by newly industrializing countries: Brazil", World Development, vol. 12, No. 5/6 (May/June 1984), pp. 575-600.

17. Petroleum Economist, December 1988.

18. Elizabeth Parsan, South-South Trading Possibilities (London, Frances Pinter, forthcoming).

19. Conjuntura, several issues, 1980-1987.

20. Petroquisa, "Produção, importação, exportação e consumo aparente dos productos quimicos e petroquimicos em 1984 e sua evolução nos ultimos dez anos" (Petroquisa, Rio de Janeiro, 1985).

21. Elizabeth Parsan, "The competitiveness issue in South-South trade: a micro-level investigation", research paper (Saint Augustine, Trinidad and Tobago, University of the West Indies, 1989).

22. European Chemical News, several issues, 1987.

23. Tarifa Brasileira Aduaneira (Rio de Janeiro, Banco do Brasil, 1984).

24. H. Fayad and Homa Motamen, The Economics of the Petrochemical Industry (London, Frances Pinter, 1986).

25. "Second world-wide study on the petrochemical industry: process of restructuring" (ID/NG.336/3).

26. Francisco Sercovich, State-owned Enterprises and Dynamic Comparative Advantage in the World Petrochemical Industry: the Case of Commodity Olefines in Brazil, Development Discussion Paper No. 96 (Cambridge, Harvard University Press, 1985).

27. Francisco Teixeira, "The political economy of technological learning in the Brazilian petrochemical industry" (Brighton, E. Sussex, University of Sussex, 1985).

28. Petro e. Quimica, November 1985.

29. Paul Collins, "Brazil in Africa: perspectives on economic cooperation among developing countries", Development Policy Review, vol. 1, No. 1 (May 1986), pp. 21-49.

30. World Bank, World Debt Tables (Washington, D.C., 1988).

MISSPECIFICATION IN THE SOURCES-OF-GROWTH ANALYSIS: CONTRIBUTION OF THE CAPITAL INPUT IN PAKISTAN*

Shahida Wizarat** and Shabbar Abbas Jaffry***

Introduction

In a paper by Wizarat [1] sources of manufacturing growth in Pakistan were identified and found to be consistent with the theoretical expectation that inputs are the major sources of growth in developing countries. Three problems related to those findings must now be considered. First, there are inconsistencies in the data on output, value added and industrial costs given in the Census of Manufacturing Industries [2]. Secondly, use of end-year capital stock might have caused misspecification of the capital input. Thirdly, a bias might have been introduced by relying on machinery prices alone to deflate capital in the form of buildings and machinery. In this paper each of those problems will be considered. Section A is concerned with cleaning of data on output, value added and industrial costs. The capital stock series for Pakistan and the capital deflators used to deflate the two components of capital input are computed in section B. In section C the sources of manufacturing growth for Pakistan are evaluated using data derived from sections A and B, and the estimates are compared with those provided in Wizarat [1]. Conclusions are presented in section D.

A. Cleaning of data on output, value added and industrial costs

A consistency check was performed on the data for output, value added and industrial cost. The consistency check involved subtracting industrial costs and value added from the value of output, or Q-S-N, with Q being output, S value added, and N industrial cost. In some years, Q-S-N was found not to equal the required value of zero.

The non-zero consistency problem was therefore resolved by the following means. First the original data source, the Census of Manufacturing Industries [2], was checked to find out whether errors had been made in data entry. Then an attempt was made to resolve

---

*Acknowledgement is due to William F. Steel, Senior Industrial Economist, World Bank, whose comments on an earlier paper contributed to this article. Thanks are also due to Professor Stephen W. Davies, University of East Anglia, and Tariq Banuri, Visiting Economist, Applied Economics Research Centre, University of Karachi, for their comments.

**Applied Economics Research Centre, University of Karachi, Pakistan.

***Portsmouth Polytechnic, Portsmouth, England.

the inconsistency by a simple transposition of digits. A value of Q-S-N equal to nine or to some multiple of nine implies that the digits have been transposed. A transposition of digits was tested as a possible solution. Each value was also compared with those obtained in preceding or succeeding years to see which was most out of line. If the inconsistency was quite small, it was ignored.

B. Revisions of capital stock data

The capital variable used in Wizarat [1] presents two difficulties. The first is the use of capital stock held at the end of the year as a measure of capital input. Capital holdings at the end of the year include capital held at the beginning of the year plus net investment (gross investment minus depreciation) during the year. Since net investment during year 1 is effective capital addition for year 2, its inclusion with capital at the beginning of the year gives effective capital for year 2.

A second difficulty concerns the use of machinery prices to deflate capital. Since capital is composed of buildings and equipment, both the components should be deflated by their respective deflators. An aggregate capital stock deflator and an adjusted capital stock series will now be computed in an attempt to resolve the above-mentioned difficulties.

1. Capital deflator

The following two capital deflators were computed: a building materials price index to deflate the proportion of investment in buildings; and a machinery price index to deflate investment in machinery.

The building material price index is a composite index of the prices of cement, pig iron, timber, paints and varnishes. Those four components were aggregated using their weights in the total building material price index from 1976 to 1984. The prices were obtained from the Central Statistical Office of Pakistan [3] for the years from 1955 to 1972, and from the Statistics Division of Pakistan [4] for the years from 1973 to 1981.

The machinery price index was obtained by multiplying the international manufacturing price index of the World Bank [5] by the exchange rates provided in the annual reports of the State Bank of Pakistan. An appropriate machinery deflator is thus obtained for all the industries, since isolated firms may be using indigenously produced capital goods. By and large, however, any aggregate large-scale manufacturing industry will import its capital goods from abroad.

2. Capital stock series

The first step towards building a capital stock series was to generate net annual investment for the years for which data are available. The following equation expresses the basic relationship:

KB + GI - DP = KE (1)

where KB = Capital stock at the beginning of the year

GI = Gross investment during the year

DP = Depreciation during the year

KE = Capital stock at the end of the year

The terms of the equation may be rewritten as follows:

GI - DP = KE - KB (2)

Net investment can therefore be estimated either as gross investment minus depreciation, or as capital stock at the end of the year minus capital stock at the beginning of the year. The identity on the left-hand side of the equation,* that is, gross investment minus depreciation, was used to generate a net investment series. Two definitions of depreciation were applied, one being actual depreciation as a percentage of beginning-year capital stock for the years 1970/71 to 1980/81. The average of those percentages were then used to compute a depreciation series. The second depreciation series was generated assuming depreciation at 5 per cent of beginning-year capital stock.

The two depreciation series were used to generate two net investment series. Net investment 1 is obtained by subtracting depreciation based on the actual average percentage of depreciation in beginning-year capital stock, while net investment 2 is gross investment minus depreciation estimated at 5 per cent of beginning-year capital stock. Net investment 2 was selected for computing the capital stock series, since it appears to be free of the sharp year-to-year fluctuations characteristic of net investment 1. The average percentage of actual depreciation, established only for the later years, does not give good results. It produces a very high depreciation rate and creates problems in generating the net investment series.

Data on the breakdown of capital stock into buildings and equipment for 1980/81 were used to divide net investment 2 into the two components. The proportion of net investment on buildings was deflated by the building materials price index, while the proportion of net investment on machinery was deflated by the machinery price index, both indexes having been computed earlier. Net investment 3 was obtained by aggregating the two deflated components of investment. Total net investment for the intervening years was generated by interpolating total net investment for those years. A merged net

---

*First the identity on the right-hand side was used to generate a net investment series. However, the series showed wild yearly fluctuations, and might yield strange results. The procedure was therefore abandoned.

investment series was then obtained by putting together the original net investment series and the interpolated values for the intervening years.

The adjusted capital stock series was computed by starting with beginning-year capital stock available for production in year 1, while investment in that year was assumed not to be productive until the following year. Net investment was then added recursively in each year from the merged net investment series obtained earlier. That gave a capital stock series for each year from 1955/56 to 1980/81. Capital stock values were then selected for years for which the other data are also available. Data used in the computation of the capital stock series came from the various issues of the Census of Manufacturing Industries of Pakistan.

C. Re-estimation and the problem of multicollinearity

Since the estimates of the sources of growth computed in this paper are to be compared with those computed in Wizarat [1], the same functional form will be used. The Cobb-Douglas production function may be written as:

$$S_t = Ae^{\lambda t} K_t^{\alpha} L_t^{\beta} \tag{3}$$

The form of the equation that may be estimated after adding the disturbance term becomes:

$$\text{Ln } S_t = \text{Ln}A + \lambda t + \alpha \text{Ln } K_t + \beta \text{Ln } L_t + \mu_t \tag{4}$$

where S = Value added in the large-scale manufacturing sector

K = Adjusted capital stock series for the large-scale manufacturing sector

L = Manufacturing employment in the large-scale manufacturing sector

t = Time trend

λ = Technical change parameter

α = Elasticity of value added with respect to capital

β = Elasticity of value added with respect to labour

μ = Disturbance term

Equation 4 was therefore estimated using ordinary least squares in the time-series-processor package. The value of R^2 was found to be very high, while the estimated coefficients had the wrong signs and were statistically insignificant. The possibility that multicollinearity might be involved was confirmed by inspecting the correlation matrix, since correlation between the explanatory variables was .9 and above. Algebraically expressed, if x is a matrix of observations on the explanatory variables, its transpose x'x will have large off-diagonal elements and relatively small diagonal elements.* A statistical technique known as ridge regression** resolves the multicollinearity problem by adding a matrix D to x'x. By adding a matrix D consisting of positive diagonal elements and zeros elsewhere, the ridge regression technique seeks to augment the main diagonal of x'x. The ridge estimate of β is therefore given by:

$$\hat{\beta} = (x'x + D)^{-1} x'y$$

In the presence of multicollinearity, $\hat{\beta}$ is more stable than the ordinary-least-squares estimate of β. Although the technique introduces a small bias in the regression coefficients, the magnitude of the bias is quite small. Moreover, in many instances it is not possible to estimate the parameters without introducing this bias.

Equation (4) was estimated using the ridge regression technique in the time-series-processor package. The estimates, along with those obtained by estimating equation (4) using ordinary least squares on unadjusted Census of Manufacturing Industries data, are presented in table 1. The two sets of estimates show that by changing the specification of the capital input and resorting to a technique that takes care of multicollinearity, a big change occurs in the estimated coefficients.

Table 1. Coefficients used in estimating sources of growth

Inputs	With unadjusted data using ordinary least squares	With adjusted data using ridge regression
Labour	.457	.906
Capital	.552	.246
Time trend	.043	.036

Note: Coefficients in the first column are from Wizarat [1]. Coefficients in the second column have been obtained by estimating equation (4) using ridge regression. The ridge regression option in the time-series-processor package does not give other statistics like the R^2, t-statistic etc.

---

*In the extreme case of perfect multicollinearity, x'x becomes a singular matrix, its determinant is zero, and its inverse does not exist.

**For details concerning ridge regression, see [6].

The estimated coefficient for labour has doubled, while that for capital has more than halved, and the estimated coefficient for time trend has declined somewhat. In Wizarat [1], the estimated values of α and β were almost 1, implying constant returns to scale, while in this paper the sum of α and β is 1.152, implying increasing returns to scale in the large-scale manufacturing sector.

The values given in the second column of table 1 were used to estimate the contribution of different sources of growth. During the period considered manufacturing value added grew at a rate of 15.12 per cent per annum. The .906 elasticity of value added with respect to labour was multiplied by the average annual rate of growth of labour input at 6.1 per cent. The contribution of labour therefore works out to be 5.5 percentage points, or 36.6 per cent (see table 2). Similarly, the contribution of capital was estimated by multiplying its growth rate of 13.9 per cent per annum by its partial elasticity, .246, yielding a capital contribution of 3.4 percentage points, or 22.6 per cent. The contribution of technological change is 3.6 percentage points, or 23.8 per cent. Moreover, α and β add up to 1.152, and the contribution of economies of scale was estimated, following Denison [7], to be 0.152/1.152 x 15.12 = 2.0 percentage points, or 13.2 per cent. All these sources of growth add up to 96.2 per cent, leaving 3.8 per cent attributable to unidentified sources.

Table 2. Estimates of the sources of growth

Inputs	With unadjusted data using ordinary least squares	With adjusted data using ridge regression
Labour	18.5	36.6
Capital	53.6	22.6
Time trend	28.7	23.8
Economies of scale	-	13.2
Residual	-	3.8
Total	100.0	100.0

Changes in the estimated values of the production coefficients have had a substantial impact on the contribution of different sources to the growth of manufacturing value added, as may be seen from table 2. Owing to a big increase in the partial elasticity with respect to labour, the contribution of labour has doubled from the 18.5 per cent estimated in Wizarat [1] to 36.6 per cent in this paper. The most startling finding is the decline in the contribution of capital from 53.6 per cent to only 22.6 per cent. This finding is highly significant, for it shows that if the capital input is correctly specified, its contribution to growth is much less than what would be found by less sophisticated methods. The contribution of technological change has declined from the previous level of 28.7 per cent to 23.8 per cent in this paper. The contribution of economies of scale was nil in the earlier paper, whereas they are now estimated to contribute 13.2 per cent to the growth process. Having quantified the contribution of inputs, technological change and economies of scale, the small remaining percentage of growth accounted for by unidentified sources may be attributed to residual factors such as improvement in skills.

D. Conclusions

Improved methods of estimating the capital input by means of a more accurate deflator and the use of ridge regression to remove multicollinearity have reduced the contribution of capital to manufacturing value added by more than a half, from 53.6 per cent to only 22.6 per cent, whereas the contribution of labour has doubled, from 18.5 per cent to 36.6 per cent. Such findings come close to those of Corea [8] for Brazil, Colombia and Honduras, Voloudakis [9] for Greece, and Lampman [10] for the Philippines, all of whom found labour to be a more important source of growth than capital. The contribution of technological change declined from 28.7 per cent to 23.8 per cent. The contribution of economies of scale to the growth process, nil in Wizarat [1], is estimated at 13.2 per cent in this paper. It must be acknowledged that a small bias might have been introduced by resorting to ridge regression, but if the alternative is no estimates at all or estimates that suffer from a small bias, then the latter is preferable.

References

1. Shahida Wizarat, "Sources of growth in Pakistan's large-scale manufacturing sector", Discussion Paper No. 88 (Karachi, University of Karachi Applied Economics Research Centre, May 1987).

2. Pakistan, Statistics Division, Census of Manufacturing Industries, various issues.

3. Pakistan, Central Statistical Office, 25 Years of Pakistan in Statistics, 1947-1972 (Karachi, 1972).

4. Pakistan, Statistics Division, 10 Years of Pakistan in Statistics, 1972-1982 (Karachi, 1983).

5. World Bank, Commodity Trade and Price Trends (Washington, D.C., 1986).

6. Ronald J. Wonnacott and Thomas H. Wonnacott, Econometrics, 2. ed. (New York, John Wiley, 1979).

7. E. F. Denison, Why Growth Rates Differ: Post-war Experience in Nine Western Countries (Washington, D.C., Brookings Institution, 1967).

8. H. Corea, "Sources of economic growth in Latin America", Southern Economic Journal, vol. XXXVII, No. 1 (1970).

9. E. Voloudakis, "Major sources of the postwar growth of the Greek economy" (Athens, 1970).

10. R. Lampman, "The sources of post-war growth in the Philippines", Philippines Economic Journal, No. 2, 1969.

SOMMAIRE

L'industrie du logiciel : les pays en développement et le marché mondial

Eoin Gahan

L'étude a pour principal objet l'environnement technico-économique qui est celui de la production de logiciel; elle passe en revue quelques-uns des facteurs qui sont à l'origine aussi bien des difficultés que connaissent les entreprises productrices de logiciel que des possibilités qui leur sont offertes. L'accent est mis sur le dosage de facteurs, l'interaction des tendances technologiques et commerciales dans le domaine du matériel et du logiciel, le rôle - positif ou négatif - de la politique industrielle dans la croissance de l'industrie, et les caractéristiques d'une stratégie efficace à l'échelon de l'entreprise.

L'étude commence par un examen des concepts et des définitions employés. Les principaux acteurs et groupes à l'oeuvre dans l'industrie, ainsi que les fonctions qu'ils remplissent, sont analysés dans la section B. Les tendances récentes dans le domaine du matériel font l'objet de la section C, l'accent étant mis sur leurs relations avec les tendances dans le domaine du logiciel, dont les principales sont étudiées dans la section D.

Enfin, les conclusions de l'étude, sous la forme de considérations stratégiques à l'intention des producteurs de logiciel, sont présentées brièvement dans la section E.

Faillite d'entreprises dans l'industrie manufacturière grecque : quelques réflexions en vue d'une politique

George H. Katsos et Joseph N. Lekakis

Les auteurs étudient les enseignements de l'échec de certaines entreprises manufacturières grecques dans un double contexte, micro et macro-économique. L'échec est conditionné à la fois par des facteurs internes à l'entreprise et des facteurs macro-économiques externes. Les facteurs internes sont rattachés au type et à la qualité de la gestion, tandis que les facteurs externes comprennent les conditions et politiques macro-économiques. Aux fins de déterminer l'importance relative des facteurs en jeu dans la

faillite des entreprises, il est procédé au calcul d'un modèle d'analyse discriminante et des tests de gestion rationnelle d'entreprises solvables et insolvables sont effectués. Les conditions qui commandent les coûts, les prix et la demande sont examinées. On conclut que les faillites d'entreprises ont résulté d'une combinaison de facteurs internes et de facteurs macro-économiques externes. De nouvelles orientations sont suggérées.

Les possibilités d'échange de biens et de services entre pays en développement : un exemple tiré de l'industrie pétrochimique

Elizabeth Parsan

L'article est consacré à l'examen des possibilités de commerce Sud-Sud dans le domaine de l'industrie pétrochimique entre deux pays se trouvant à différents degrés de développement, le Brésil et Trinité-et-Tobago. Il ressort d'une analyse de leurs ressources que la Trinité-et-Tobago pourrait exporter certains produits pétrochimiques au Brésil, lequel pourrait fournir de la technologie destinée au développement de l'industrie pétrochimique trinidanienne. Des liens commerciaux de ce type devraient faciliter l'expansion des échanges et assurer d'importants avantages, statiques comme dynamiques, aux deux partenaires. Cependant, il est relevé que les obstacles existants, notamment les barrières tarifaires et non tarifaires, ont entravé le développement du commerce. Diverses mesures d'ordre politique sont par conséquent suggérées pour éliminer ces obstacles.

Estimation des sources de croissance : la contribution des apports de capital au Pakistan

Shahida Wizarat et Shabbar Abbas Jaffry

Il ressort de l'analyse présentée dans l'article qu'une spécification plus précise des apports de capital et l'emploi d'une méthode qui prend en compte la colinéarité, permettraient de réduire de plus de moitié la contribution des apports de capital à la croissance de la valeur ajoutée dans l'industrie manufacturière, et de doubler, ou presque, celle du travail. Le modèle perfectionné, appliqué au Pakistan, attribue pour une part la croissance de la valeur ajoutée aux économies d'échelle, à la différence du modèle classique dans lequel la contribution de cette source est égale à zéro.

EXTRACTO

La industria del "software": los países en desarrollo y el mercado mundial

Eoin Gahan

Este estudio trata de las condiciones tecnoeconómicas en que tiene lugar la producción de "software", y contiene un análisis de algunas de las fuerzas que en esta industria crean dificultades o brindan oportunidades a las empresas. Se concede atención especial a la combinación de fuerzas, a la interacción entre las tendencias, tanto tecnológicas como comerciales, del "hardware" y del "software", al papel desempeñado por la política industrial como favorecedora u obstaculizadora del crecimiento de la industria y en las características de una estrategia eficaz a nivel de empresa.

El estudio se inicia con un examen de conceptos y definiciones. En la sección B se analizan los agentes básicos, los grupos que actúan en la industria y las funciones que ejercen. La sección C trata de las tendencias recientes del "hardware", con especial referencia a su relación con las tendencias del "software", las más importantes de las cuales se examinan en la sección D.

El estudio termina con un resumen de conclusiones, en forma de consideraciones estratégicas destinadas a los fabricantes de "software", que figura en la sección E.

El fracaso de las empresas en la industria manufacturera de Grecia: algunas pruebas para la adopción de políticas

George H. Katsos y Joseph N. Lekakis

En este documento se examina, en el doble marco de la micro y la macroeconomía, la experiencia de empresas manufactureras griegas que fracasaron. El fracaso es consecuencia tanto de factores internos de la empresa como de factores externos agregados. Los primeros tienen que ver con el tipo y la calidad de la gestión, mientras que los segundos se refieren a las condiciones y políticas macroeconómicas. Con el fin de calibrar la importancia relativa de los factores que intervienen en el fracaso de las empresas, se evalúa un modelo de análisis discriminatorio y se realizan pruebas de gestión eficaz en empresas solventes e insolventes. Se examinan las condiciones en materia de costos, precios y demanda. La conclusión es que los fracasos de las empresas se deben a una combinación de factores internos de la empresa y de factores externos agregados, y se sugieren cambios de política.

El potencial de comercio de bienes y servicios entre países en desarrollo: un ejemplo de la industria petroquímica

Elizabeth Parsan

En este documento se examina el potencial del comercio Sur-Sur en la industria petroquímica, tomando como ejemplo dos países que se encuentran en distintas fases de desarrollo: Brasil y Trinidad y Tabago. El análisis de los recursos de este último permite ver que tiene posibilidades de exportar al Brasil determinados productos petroquímicos, y que el Brasil, a su vez, podría proporcionar a Trinidad y Tabago tecnología para el desarrollo de su industria petroquímica. Cabe esperar que tales relaciones comerciales faciliten la expansión del comercio y produzcan importantes beneficios, tanto estáticos como dinámicos, a ambos interlocutores comerciales. Se hace notar, sin embargo, que existen obstáculos, especialmente las barreras arancelarias y no arancelarias, que vienen dificultando el desarrollo del comercio. En vista de ello, se sugieren varias medidas de política para eliminar tales obstáculos.

Estimación de las fuentes de crecimiento: la contribución del insumo de capital en el Pakistán

Shahida Wizarat y Shabbar Abbas Jaffry

El análisis presentado en este documento muestra que, especificando mejor los insumos de capital necesarios, y utilizando una metodología que tenga en cuenta la colinealidad, se reduce en más de la mitad la contribución de los insumos de capital al crecimiento del valor añadido manufacturero, mientras que casi se dobla la contribución de la mano de obra. El modelo mejorado, aplicado al Pakistán, atribuye cierta proporción de crecimiento del valor añadido a las economías de escala, a diferencia del modelo convencional según el cual la contribución de las mismas es nula.

INFORMATION FOR CONTRIBUTORS

The Supervisory Panel of *Industry and Development* welcomes contributions relevant to the stated aims and scope of the journal. Authors should contact the Supervisory Panel at the address given below.

1. Articles may be submitted in English, French or Spanish to:

   Supervisory Panel of *Industry and Development*
   Studies and Research Division, UNIDO
   P.O. Box 300
   A-1400 Vienna, Austria

2. Articles submitted should contain original, unpublished work not being considered for publication elsewhere.

3. The Supervisory Panel does not accept responsibility for loss of manuscripts.

4. Two copies of the manuscript should be submitted. It should be typewritten in double spacing on white bond paper, and all pages should be numbered consecutively.

5. The first page of the manuscript should contain the title, the name and institutional affiliation of the author, a summary of the article not exceeding 100 words, and the current address of the person to whom proofs may be sent.

6. Mathematical expressions should be numbered throughout the manuscript. If a derivation has been abbreviated, the full derivation should be presented on a separate sheet, which will not be published. The use of complex mathematics should be kept to a minimum.

7. Footnotes should be numbered consecutively throughout the text. References should include complete bibliographical data: author, full title of paper (or book), publisher, and place and date of publication. A reference to an article published in another journal should include that journal's full title, as well as the place and date of publication, author, volume number, issue number and page numbers. See a current issue of *Industry and Development* for the style employed for footnotes and references.

8. Illustrations and tables should be numbered consecutively throughout the text and provided with clear captions and labels.

9. Fifty offprints of their articles will be made available to authors.

10. Authors should note that, in accordance with UNIDO publications policy, articles accepted for publication are subject to editorial review by the UNIDO Secretariat.

Printed in Austria
V.89-59431—March 1990—1,900

02400P

UNIDO publication
Sales No.: E.90.III.E.10
ISBN 92-1-106248-9
ISSN 0250-7935

ID/SER.M/28