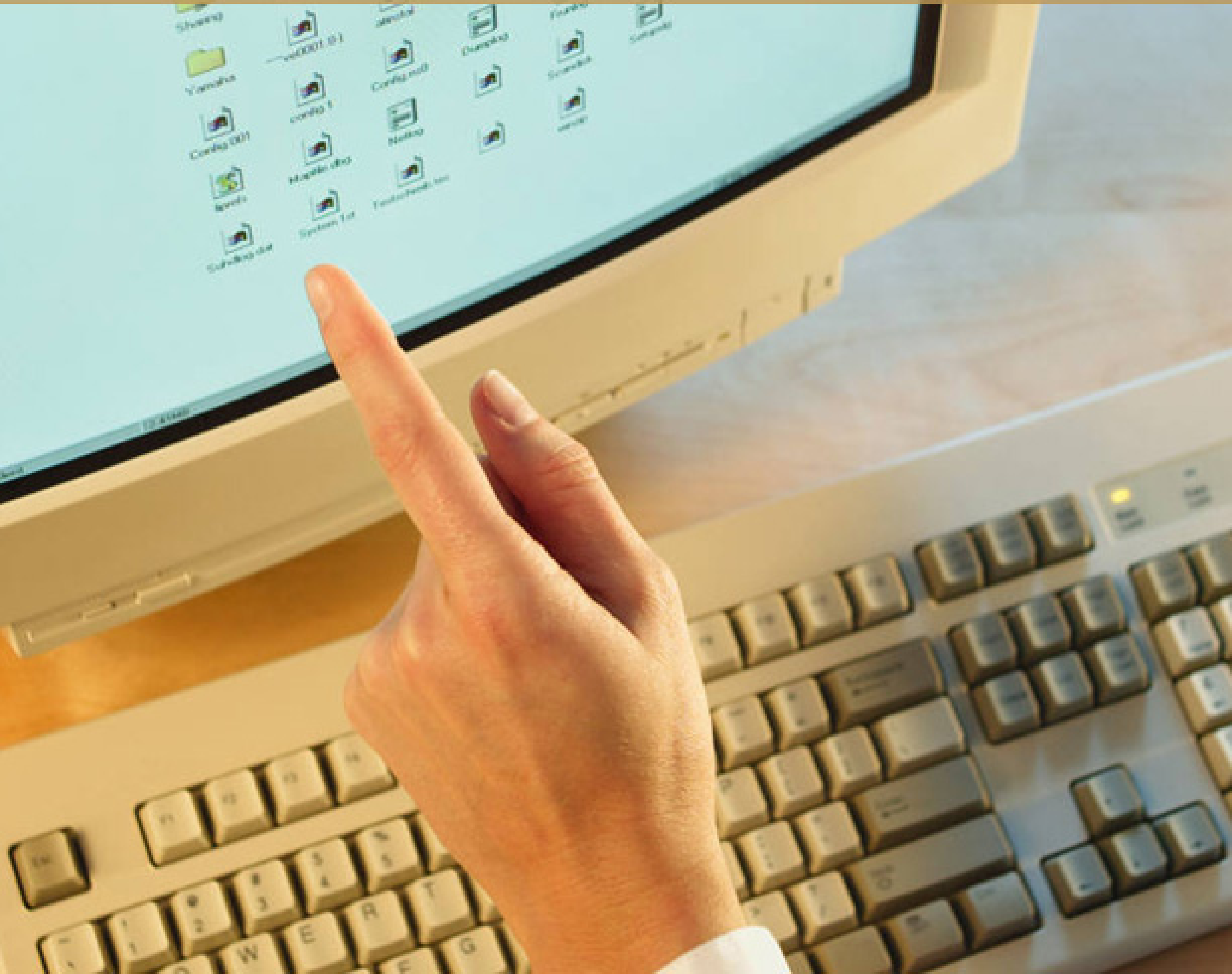
Teacher's Handbook of Information Technology

Ralph Forrestal



First Edition, 2009

ISBN 978 93 80168 64 7

Published by:

Global Media
1819, Bhagirath Palace,
Chandni Chowk, Delhi-110 006
Email: globalmedia@dkpd.com

Table of Contents

One

INTRODUCTION

Information is the life-blood of a community information service and the information file is at its heart. Unless the heart is sound and continually pumping a supply of regularly renewed and fresh information into the system, it will not function at its best. Therefore, it is important to give extra care and attention to planning the resources needed to set up a sound information base and a workable system for keeping it up to date.

Some basic things you will need to consider are as follows:

— How is the information to be collected?

— What method is to be used for processing and storing information?

— How is the information to be retrieved?

— What size is the file likely to be?

— What areas, geographical and subject, is the file to cover?

In addition, you will need to consider how the information is going to be disseminated, if it is not just for internal use. The kind of system you choose and the degree of its sophistication will depend on the nature of your information service, the size and complexity of the information you need to store, the staff, and the financial resources available. When designing a system, make sure it is readily understandable to all those using it-volunteers and public included-and not just to professionals.

NOBLE INFORMATION

It is highly unlikely and undesirable that an information service should need to 'go it alone' in the collection of information. Such is the volume, complexity and variety of sources of information in present-day society that, unless your service has very narrow terms of reference, it will be virtually impossible for it to collect and keep up to date all the information that is required.

Therefore, it is important that you first of all identify the information providers, support services and 'gatekeepers' in your community and establish effective contacts with them. These links may already have been forged in the process of conducting a community profile. They now need to be fostered and strengthened, so that there can be a mutual exchange of information. These contacts will also be able to provide you with useful feedback from the community as to the success or otherwise of your service. The network of contacts can be maintained on a fairly casual basis as the need arises, or you might want to formalize the arrangement. Some ways of doing this include irregular 'get togethers', informal luncheon clubs, regular meetings with agendas and minutes, or the circulation of a newsletter or bulletin. Other activities that could develop out of such meetings are joint collection of information, shared publicity, compilation and publication of directories, information handbooks and leaflets, training, and general discussion of common problems. Collecting information is a time-consuming process and there is no one method of going about it. You will probably have to use a combination of several techniques to build up a satisfactory information base. One thing definitely to be avoided is duplicating work that has already been done. So first of all identify the following:

Existing information files. Contact all the other community information services, council departments and Organisations who are likely to maintain information files and ask if they would be willing to share this information. Try to offer something in return, either an exchange of information or some other help you can provide.

Local directories. There are several types of directories that may be available in your area and which are useful sources of soft information: (a) local government authorities often produce directories or town guides of their area which contain a certain amount of local information or they may produce a guide to council services; (b) telephone directories, especially Yellow pages which have an alphabetical subject arrangement; (c) area directories, such as those produced by Thompson's Newspapers, which are similar to Yellow pages but also contain a section of community information-your local newspaper may produce a directory of local services, sometimes in conjunction with local information groups; (d) citizen's guides may also be produced by the local newspaper as a supplement to an existing newspaper; (e) CVSs or RCCs sometimes produce directories of voluntary Organisations in their area or you may find these listed in your local CVS/RCC annual report; (f) directories covering a special subject, such as accommodation, halls for hire, etc.; (g) directories aimed at a particular interest group or groups e.g. people with disabilities or senior citizens.

National directories are useful not only for the wider network of services but also for identifying local offices of a national Organisation.

Other methods to use in building up your information base and maintaining its currency include:

Looking and listening: A lot of information can be gathered by simply walking around your community, looking at notice-boards, picking up leaflets, or attending open meetings, fairs, fetes and other community events. This method of collecting information is unpredictable and requires more than a 9-to-5, five-day week.

Contact with individuals: People are a major source of information in any community and there are always those to whom others naturally go for information. Try to identify such people and enlist their help, possibly by offering something in return-bartering is a time-honoured system! Direct contact with individuals in other information services and Organisations is often a better way of eliciting information than impersonal methods and also paves the way for further cooperation. Pay regular visits to such centres.

Scanning newspapers, magazines, newsletters, etc., can turn up details of new Organisations; changes of personnel, premises or hours of opening; new services introduced; and may highlight problem areas where information may be needed.

Other documents, including council minutes and agendas, planning regulations, annual reports (ask to be put on mailing lists for these), leaflets and manifestos.

Media publicity Local press, radio and cable television stations may be prepared without charge to put out a call for Organisations to contact your service. If not, then consider inserting an advert or, where there is, a local directory produced, have a tear-out slip included for Organisations to fill in and return to you. As a result of these various methods, an amount of raw material will have been obtained which may be of use to your information service. How accurate, up to date and complete the data is will depend very much on the reliability of the source. In most cases, unless you have every confidence in the source, the information will have to be checked before it is entered in a more permanent form in your resource file. So, at this stage, simply record it onto say a scrap 5in x 3in card and put it into a temporary file.

The information you have obtained will be of several different kinds but the most common will probably be that relating to Organisations, clubs and societies. This is the most difficult type of information to collect since it is ever changing and therefore unrelenting effort will be required to achieve even reasonable completeness or accuracy-100% will be impossible. However, with a good system for collecting and updating the information, you should be able to build up and maintain an acceptable file.

A lot of staff time will be needed to carry out this work, so first of all investigate whether there are other local information services that need this type of information and would be willing to help in its collection. It may be possible to collect the information through an umbrella group, where the work can be framed out so that it is not too great a burden on any one Organisation. Alternatively, a joint approach can be made on behalf of a number of Organisations.

Ideally, the most effective way to collect information of this kind is to pay a personal visit to each Organisation or service, but rarely will that be possible without extensive staff resources or a very small community. The cheapest and quickest way to collect this information is by telephoning. Prepare a standard list of questions to ask so that entries in your information file follow the same format.

Telephoning is only practicable if you have a reasonably small number of Organisations to contact, say up to one hundred. If visits and telephoning are ruled out, then the information can be collected initially by postal questionnaire, with a request for up-to-date information sent at least once a year and also when it is known from other sources that changes have taken place. A covering letter should be sent with the questionnaire or update request, explaining "Why the information is needed and asking for cooperation. If the information is to be passed on to other Organisations or commercial bodies, either free or for payment, this should be drawn to the attention of the recipient of the questionnaire, so that they have an opportunity to decline to give the information or insist that it is not so disseminated. If the information is to be stored in a computer, then you will need to meet the requirements of the Data Protection Act.

If it can be afforded, include with your questionnaire some means of returning it free of charge, as this will significantly improve the response rate. Separate forms may be required to collect different kinds of information. The following is a suggested list of headings which you might need to include on forms for collecting information from two of the most common types of Organisations.

Clubs and Societies

— Name: the name by which the club is best known, plus the full name if different. Indicate relationship to larger body, i.e. branch, regiment, lodge, etc.

— Secretary's name, address and telephone number or those of a similar person if the Organisation has no secretary.

— Place, day and time of meetings.

— Purpose of Organisation if not self-evident from the name.

— Eligibility: any restrictions by age, sex, ethnic group, occupation, status (single, divorced, widowed) etc.

— Subscription: any charge to become a member or to attend meetings.

— Annual general meeting: the date of an Organisation's AGM can be a useful indication of when to send out update forms as this is when officers or policies are likely to change.

— Other officers: names, addresses, telephone numbers of treasurer, president, chairman, publicity officer, membership secretary, etc.

— History of the Organisation including special events or persons associated with it.
— Publications: newsletter, diary, commemorative brochures, annual report, etc.
— Date the information was obtained and/or last updated or checked.

Agencies and Organisations providing a service

— Name: popular name and full name, e.g. CHAT (Come Here and Talk), SHAC (Shelter Housing Action Centre), plus relationship to parent body where necessary.
— Address: street address and postal address, Post Office, box number if used.
— Telephone and fax number, including-service number, administration number, after-hours or hotline numbers.
— Contact person: personal name, title and address, if different from above.
— Hours of opening: days and times; note if seasonal, e.g. holiday period, term time.
— Services provided: advice, counselling, practical help, etc.; types of enquiry the service would like to have referred to them.
— Eligibility: age, income, sex, residency, status.
— Application procedures: walk-in, appointment or waiting list; what papers or documents need to be brought.
— Cost: free, fees, means-tested, donations; any facilities for payments to be spread over a period; help available from local or central government, charities, etc.
— Geographical area served: neighbourhood, city, county, region, ad hoc area; no geographical restrictions.
— Branch offices: extension bureaux, mobiles, surgeries, etc.; include hours of opening, routes and times of stops.

— Director, administrator or executive director of the service; name and telephone or extension.

— Volunteers: does the service use volunteers and for what purposes? Method of recruitment.

— Publications: directories, handbooks, leaflets, annual report, etc.

— Funding: local government grant-aided, donations, etc.

— Special facilities: foreign languages spoken, access and facilities for people with disabilities, photocopying or fax service, escort, advocacy, etc.

— Transportation: how to get to the service, e.g. bus route numbers, underground line and station, etc.

— Date information was obtained and/or last updated or checked.

These two lists represent most of the facts that a community information service might need to know about an Organisation or service. In practice, depending on the scope of your service, it may not be necessary to include all the fields in the questionnaire. It is a good idea to get the person filling in the questionnaire-to-sign it, thereby giving the information service some-protection against claims of giving out wrong information—a not uncommon occurrence, especially when the information is committed to print.

There are other types of 'soft' information that your service might need to collect, for example halls for hire, places of local worship, accommodation, 'What's-on' events, local industry. The procedure is much the same:

— Decide what information you need to know.

— Devise a standard format or list of questions to ask.

— Identify possible sources of information.

— Decide which method or methods to use to collect the information: telephone, personal visit, postal questionnaire, press advert, etc.

How to Store 'soft' Information?

There are various methods that can be used for storing information:

A list is easy and quick to consult, can be photocopied for clients to take away and can be faxed, but has very little flexibility for inserting and updating information.

A strip index provides flexibility, can easily be photocopied, and is available in various forms: large binders holding many pages, smaller folders containing just a few pages, address books, wall-hung panels, rotunda units, etc. Strips are available in several colours, which allows for a simple categorization, and various widths. Even so, they are limited as to the amount of information that can be contained on them. It is also not quite as easy to insert new strips as the manufacturers claim.

Cards are still one of the most commonly used systems for storing information. They are infinitely flexible, easy to insert and update, can be sorted into any preferred order, and require very little technical know-how. The most popular sizes are 5in x 3in, used mainly for indexes or temporary information waiting to be checked, 6in x 4in, and 8in x 5in, which are large enough to contain sufficient information about most Organisations or services and to be pre-printed with a standard grid for recording the information. There are more superior types of card files available, such as the Rondofile or Rotadex systems which use custom-designed stationery and therefore tend to be more expensive. With systems that use standard cards, you can-if you are hard-up or want to save trees-use the reverse of scrap cards. Card files have a number of disadvantages: they are not so easy to carry around, they arc not easily reproducible if someone wants a copy of a particular section, multiple access (e.g. by name, subject, area, service, etc.) requires duplication of cards, and the updating process can be tedious.

You may still find around edge-punched cards which offer a primitive type of multi-access to the information on the card-through the strategic use of knitting needles or rods. Personally speaking, I would save up and get a computer.

Loose-leaf binders with one sheet for each entry, sometimes called 'sheaf' binders, have a similar flexibility to cards. They are slightly slower to add to and amend but are more portable. They are suited best for alphabetical arrangements as they are difficult to guide.

Microcomputers now have the capacity to handle immense information files that can be accessed by any number of predetermined variables or even free-text searching by keyword, depending on the software used. With a printer attached, all the information or sections of it can be printed out on demand or to provide a master for printing by conventional methods. Personal computers now begin at prices that bring them within the range of most community information services. However, there is danger in automatically seeing the computer as the best solution for maintaining your information file, as this may not always be the case. A lot depends on the size of your file and its complexity, what you want to do with the file, and to what other uses you want to put the computer. The importance of the computer in, information and advice work is such that it merits a chapter to itself, and you will find these and other aspects.

Filing System

Whichever method is chosen for physically storing the information, except for that by microcomputer, it will be necessary to decide the best way of arranging entries so that the information can be retrieved swiftly and accurately. In order to ensure that the total resources of your information service are used to the full when answering an enquiry, it will help if the system used to organize the information file can be integrated with that for the vertical file and the book-stock. This point needs to be borne in mind now, although it will be dealt with more fully later.

The bulk of the information file is likely to comprise 'soft' information -which you have collected about Organisations, clubs, societies and services and, to a lesser extent, about individuals who are a valuable source of information or advice, plus items of 'hard' information not available or conveniently available in printed form. There are several ways in which you might want to retrieve this information-by the names of the Organisations, by subject (i.e. area of interest), by clientele, by the type of service (e.g. counselling), or by place. In practice, the 'clientele' and 'type of service' approaches are usually catered for through the subject file rather than, as separate files.

Organisations file: The master file for the whole system will usually be an alphabetically arranged sequence of entries by name of Organisation. It is generally recommended that the full, official name of the Organisation is used, with references from other forms of the name. In practice, it does not really matter as long as there are references from the alternatives to whichever form is chosen. There is a good argument for putting the main entry for an Organisation under the form of name by which most people refer to it, even if this differs from or is an abbreviation of the official name. This is especially so where the file is to be used by the public.

In addition to details of Organisations, each entry in the master file may also contain certain 'housekeeping' details relating to the maintenance of all the information files, plus references to appropriate headings in the subject file, where relevant information can be found, and a classification number.

Subject file: Most enquiries received by a community information service are likely to be about the need for a service or activity or for help with solving a problem, rather than for information on a specific named Organisation. So the information you have collected about Organisations, services and individuals will also need to be accessible by the services and activities they provide. There are two ways this can be achieved, by either using a subject index or having a subject file of Organisations. In both, you will need to choose or adopt a set of subject headings which will adequately describe the information on file or elsewhere in the system and the interests of your clientele.

If you are compiling your own list of subject headings, choose terms that are in common use by your clientele, especially if they will be consulting the files directly, rather than the 'official' term, e.g. OUT OF WORK or ON THE DOLE instead of REDUNDANT. Always refer from alternatives not used to the chosen term.

A subject index is rather like the index to a book. It is an alphabetically arranged file of subject heading cards on which reference is made to where information on that subject can be found. The cards themselves do not contain information. The subject index can contain references to Organisations, individuals and other supplementary material such as pamphlets, periodical articles, audiovisual material, books, etc.

A subject file consists of an alphabetically arranged set of subject heading cards behind which are filed copies of the Organisation cards appropriate to that heading and possibly cards containing items of 'hard' information. An alternative is to arrange cards according to a classification system either devised by yourself or adopted from another service, such as the NACAB classification scheme.

Place file: Where an information service covers a number of distinct towns, villages or neighbourhoods, each having a similar range of Organisations, it may be helpful to have a file that can be accessed by place. There are a number of options. You could sort the Organisations file initially by place, especially if it contains few Organisations whose responsibilities extend to the whole area covered by your service. Alternatively, if most of the Organisations are prefixed with the name of the place (e.g. EXVILLE ATHLETIC CLUB, EXVILLE COUNCIL FOR THE DISABLED), the alphabetically arranged master file will automatically bring together those from the same location, provided inverted headings have not been used.

Some arrangements will have to be made for Organisations which do not conform to the pattern, either because the place-name does not feature at the beginning or because it is not contained in the name of the Organisation at all. In these cases, an additional card can be made out with either an inverted heading (e.g. EXVILLE, ROTARY CLUB OF) or the place-name can be prefixed. The third option is to have a separate file of Organisation cards arranged initially by place and then alphabetically by name of Organisation or by subject.

Filing Alphabetically

Filing alphabetically is not quite as simple as ABC, as any librarian will tell you. There are two recognized methods, known as 'word by word' and 'letter by letter'. In word by word, entries are initially sorted by the first word in the heading, then headings that begin with the same word are sorted by the second word, and so on. Another way to describe this method, which may be more helpful, is to treat spaces between words in a heading as an imaginary letter coming before 'a' in the alphabet, then sort letter by letter.

In letter by letter, you simply ignore any spaces in the heading and sort one letter at a time from left to right. Here is how a small group of headings would look sorted by the two methods:

Word by word	***Letter by letter***
DO SOMETHING! BOYS CLUB	'DOG AND BONE' DARTS TEAM
'DOG AND BONE' DARTS TEAM	DOGSBURY RESIDENTS' ASSOCIATION
DOG SNIFFERS ANONYMOUS	DOGS HOMES
DOGS HOMES	DOG SNIFFERS ANONYMOUS
DOGSBURY RESIDENTS' ASSOCIATION	DO SOMETHING! BOYS CLUB

There are several other niceties to do with filing alphabetically but we only need mention here that hyphens are treated as spaces in word-by-word sorting and numerals are spelt out as said, so that 1900 (the year) is considered for sorting as Nineteen Hundred and 1,900 as One Thousand, Nine Hundred.

File 'housekeeping'

In addition to the information needed to answer enquiries, there are other items of information to do with the maintenance of the file which may usefully be included on each entry, such as the following:

— The date the information was obtained, last updated or last checked, which will indicate not only the degree of reliability to be placed on it but also when to update;

— The date a questionnaire or update letter was sent - a check for chasing up non-returned forms;

— Additional contacts or other information which does not fit into the main part of the entry;

— Subject headings used in the subject file;

— Place headings used in the place file, if not obvious from the name of the Organisation;

— Feedback-comments from users of the service;

— 'Tracings', i.e. a list of other cards that refer to they can be traced and amended or withdrawn when necessary.

The amount of this 'housekeeping' information you need to include will depend on how sophisticated you need your file to be. The bigger the information service and the larger the file, then the more likely it is that you will need to introduce a systematic procedure for processing information. In such a case it is usual for the housekeeping information to go on the reverse of each entry, using a standard grid like the one illustrated here. It is arguable that if your information file requires this degree of sophistication, you ought seriously to consider keeping it on computer.

For a small community information service, it may be quite adequate just to note the date when the information was collected, checked or updated.

Updating

By their very nature the entries in your database will be changing continually. Hours of opening, meeting places, officers, membership fees, subscriptions, charges, etc., are all susceptible to frequent change. Therefore, it is important to have a system for regularly updating each entry in the database. There are several ways of achieving this but, whichever method you adopt, it must be regular and ongoing.

Interim Updating

All the entries for Organisations should be checked at least once a year to ensure they are still correct. In between times, however, new information will be brought to your attention by various means, including word of mouth, newspaper reports, Organisation newsletters, company reports and by direct contact with the Organisation itself. If the source of information is reliable, it can be substituted immediately for the out-of-date information in the database but, if not, it should be noted and a further check made to verify its accuracy. A simple way to add new information to an entry in a card file is to write it onto self-adhesive labels or slips (obtainable at most stationers) which can be stuck over the original information. Even though an entry has been updated in between times, it should still be checked formally once a year.

Annual Updating

Once a year is probably a reasonable time-span for updating your database but a shorter interval may be necessary if the information content changes frequently. You will need to decide how the annual update is to be carried out. Some Organisations like to update the entire file at the same time, so that they can say the file was as accurate as possible at a particular date. However, this does create a tremendous amount of extra work for a period of several weeks and you may feel it necessary to get extra paid, work-experience or volunteer help. Another disadvantage of this method is that it captures a picture of Organisations at a set moment in time. Some Organisations may be just about to make changes, and your records for these would be out of date for almost a year.

Alternatively, you may decide to update on a continuous basis by using either the date the record was added to the database or the Organisation's annual general meeting (AGM) date as a trigger. This is easy to do if your database is kept on a computer, since a date field can be used to produce a subset of records and, by linking the database with a word processing package, a personalized standard letter can be printed out for sending to each Organisation. It is less easy with a card or any other kind of file. You can get little coloured metal tags to clip on to the top edge of cards, a different colour for each month, or you could use an appropriate symbol on the heading of the card, such as a coloured dot, a letter or number. Tags can make the file look a bit messy and are easily knocked off if it is in constant use; symbols require the whole file to be checked through.

It is better to update records shortly before or after each Organisation's AGM, since most Organisations, if they are going to change their officers or constitution, generally do it once a year at the AGM. This is so because many Organisations do not know their AGM date a year in advance, it is enough just to indicate the month in which the AGM is usually held. For those Organisations and services which do not have an AGM, you will need to use the date the information was added as the trigger.

When writing to Organisations to update their information, it is better to send them a copy of their record with a request to indicate any changes, rather than sending a blank questionnaire. This saves the secretary unnecessary time in repeating information that has not altered.

An alternative to writing to each Organisation is to phone, but this is time-consuming and costly, and only feasible if the number of Organisations to contact is fairly small. Avoid sending out updates at times when people are likely to be away, such as during the summer. There are certain times when it is more appropriate to update other kinds of information, e.g. social security benefit rates (which usually change in April and September), adult education classes or school, college and university courses which change term-wise.

When an updated entry is returned it should be checked to see if the subject or place headings still apply. Minor changes can be recorded on the master card and any other cards in subject or place files; major changes may require a new set of cards to be produced. For those with computerized databases the process is simpler as just the one record has to be updated. Don't forget to change the date the information was last checked or updated.

OUTSIDE INFORMATION

As well as the database, most information services will need a certain amount of 'hard' information to answer enquiries. Short, unrecorded or inadequately recorded items of 'hard' information, as we have already seen, can be incorporated into the database. However, the majority of hard information will usually be found in one or more of the vast range of print forms, starting from locally produced free broadsheets to extremely expensive multi-volumed loose-leaf reference works.

Within the scope of this book, it is not possible to go into detail about the sources of this material, since they will vary considerably according to the type and subject range of the service you are operating. Community information changes rapidly so any printed sources should be treated with care. However, you might find useful a chapter on the sources of community information that I contributed to Printed reference material and related sources of information, 3rd ed., Library Association Publishing, 1990, which concentrated on Organisations and on those items which are regularly updated.

The following framework for collecting community information was originally devised by Grainne Morby, who at the time was working for the Community information Project, with some additions of my own:

Distinguish between subject areas

In other words, identify the main topics into which the subject scope of your information service naturally divides. Throughout this framework I will take as an example an information service for which the main area of interest is housing. It is sometimes useful to apply a consistent criterion when dividing up a subject, although obviously this would not be so easy for an information service whose scope covers as broad an area as, say, 'community information'.

However, taking housing as our example, you could decide to divide it by type of accommodation, e.g. owner-occupied, private-rented, council-rented, New Town, Housing Association, institutional accommodation, tied accommodation, mobile homes, etc. In most cases you would probably need an additional category for subjects that cut across all or more than one category. In the case of housing, this might be 'squatting' or 'homelessness'.

Since the material you want to collect will usually be written to meet a particular need or needs, it is advisable to collect information around those needs rather than for fitting into a theoretical framework. The kind of broad client groups that might be identified in the field of housing are landlords, tenants, owner-occupiers, squatters, transients, elderly people, people with disabilities, the homeless. Then, one could identify client groups who share a common problem or need associated with their particular type of accommodation, such as eviction, dampness, house repair, renovation, harassment, mortgages, rent arrears, planning permission, redevelopment, etc. Once these subject areas and client needs have been identified, it should give you clearer guidelines when you come to select material from the various sources identified in the next part of the framework.

Print sources

1. Official
   - (a) National
     - — *pre-legislation:* political party manifestos and policy documents, speeches by Ministers, the Queen's Speech at opening of Parliament, White Papers, Green Papers.
     - — *Legislation:* Bills, Hansard reports of debates in both Houses of Parliament, Acts, Statutory Instruments.
     - — *Guidance to local statutory bodies:* circulars and letters from Ministers, guidance notes, reports of Inspectors.
     - — *Reference books:* collections of legislation, e.g. Statutes in force, Legal aid handbook.
     - — *Periodicals:* from government departments and agencies, e.g. Department for Employment gazette.
     - — *Guidance to the public:* leaflets, posters, videos and other A/V material.
     - — *Ombudsmen*.
   - (b) Regional
     - — Health authorities, British Rail.

— Privatized utilities: gas, water, electricity, telephone.

— Consumer watchdog bodies: Post Office Users' National Council (POUNC), Office of Gas Supply (OFGAS), Office of Electricity Regulation (OFFER), Office of Telecommunications (OfTel), Office Of Water Services (OFWAT).

(c) Local

— *Local authority policies and decisions:* agendas, minutes, standing orders, bylaws and regulations.

— *Guidance to the public:* handbooks, leaflets, etc. from local authorities, area and district health authorities, Community Health Councils, local offices of government agencies.

2. Non-official

(a) National

— Reference books and handbooks from commercial publishers.

— Voluntary Organisations.

— Pressure groups, self-help groups.

— 'Umbrella' Organisations, e.g. National Council for Voluntary Organisations, Federation of Independent Advice Centres.

— Professional bodies, e.g. The Law Society.

— Trade associations, e.g. National Association of Retail Furnishers.

— Educational bodies, e.g. National Extension College, Open University.

— Trade unions.

— The media.

— Practitioners in the field who publish.

(b) Local

— Claimants' unions.

— Law centres, other specialist advice centres.

— Neighbourhood and generalist advice centres, resource centres.

— Local 'umbrella' groups.

— Pressure groups, campaigning groups, action centres. Voluntary Organisations.

— Local media: newspapers, community newspapers, radio, cable television.

(c) New technology

— Video-text systems: Prestel, teletext, private viewdata.

— Online computer databases, e.g. Volnet

— CD-ROM, videodiscs, etc.

This list is not intended to be exhaustive but will give you some idea of the range of sources in which to look for information.

Howard Matthews, in his book *Community Information* has drawn up a set of criteria for selecting community information materials. It is aimed particularly at library community information services but may be generally helpful for anyone setting up such a service.

1. Selection should be done by those who know the community best.
2. Material should relate to specific, local, identified needs.
3. All material should be current.
4. All material should be written at a level appropriate to its intended use.
5. Material should be concerned with the practicalities of problem solving.

6. All new material should either fill a gap, offer a new viewpoint, or promise something better.
7. Material should take into account the level of use the client will make of it.

Obtaining publications and other material

Once the material you require has been identified, you will need to obtain it. Some material will be priced, in which case- you will need to order it, and some will be free.

Priced publications

The procedure for obtaining priced publications, particularly in the field of community information, is not as simple as you might suppose. Since a lot of the material may well be fairly cheap and the small or community publishers, booksellers and library suppliers are reluctant to obtain it as the profit margin is likely to be too low for the amount of effort involved.

The alternative is to order direct but this also has its difficulties as many small publishers now insist on money with order, and this can cause problems for some Organisations through the need to raise a cheque or limitations on the amount that can be spent from petty cash.

A few national voluntary Organisations, e.g. Age Concern England, operate a standing-order system for their publications-this is one solution, although you may need to accept blanket coverage of the Organisation's output. In the larger cities, you may find a community or alternative bookseller who is prepared to supply this material to order. A directory of alternative bookshops is published annually by the magazine *Radical Bookseller*.

Whatever the method of ordering, it is important to keep a record of what has been placed on order and from whom to avoid duplication of titles and as a check on supply. For each title make out an order or slip. This can be either a scrap

5in x 3in card or one specially printed with a standard grid. The following indicates the kinds of information that need to be recorded:

— Title of publication-more important than author for this kind of material, since authorship is often unclear;
— Author-might be an individual or an Organisation;
— Price, date of publication and, if a periodical, frequency;
— Supplier-this may be a bookseller, library supplier or the address of the producing body;
— Date of order;
— Source of information about the item;
— Location, if you are ordering for more than one centre.

You may find it helpful to use the order cards later as the basis of a catalogue of the publications in your information centre; in which case., leave space at the top of the card for either a subject heading, classification number or filing code. After an order has been placed with a supplier, the order cards are filed, preferably by title, in one alphabetical sequence known as the 'order file'. When a publication is received, the order card is removed from the file and accompanies the publication to the next stage if it is to be used later as the catalogue card. If not, you may find it useful to file the order card in a 'publications received' file, until such time as the publication is permanently recorded in the system, after which the card can be thrown away.

Periodicals

Publications which arrive at regular intervals—weekly, fortnightly, monthly, quarterly—will not require an order card but you will need to keep a record of each title taken. This should include the following items of information:

— Title of periodical;
— Frequency;

— supplier;
— when subscription is due for payment;
— a grid to record receipt of each issue;
— instructions for disposal of back copies.

For those using computers, there are software packages available for periodicals management but these are aimed at Organisations who take an extensive range of periodicals and such packages are not cheap. If you have a database management package on your computer, you would do better to devise your own periodicals database.

Free material

This can vary from quite substantial loose-leaf binders plus updates to leaflets, posters and bookmarks and may be required in bulk for clients to take. Since this is likely to be a frequent type of request, it may be quicker and simpler to duplicate a standard letter to send to Organisations, with space left to fill in address, title of item, format (leaflet, booklet, poster, etc.) and number of copies required. It is advisable to keep a record of what is requested, if only as a check on whether it has been received or not.

A lot of the sweat has been taken out of identifying free materials, and addresses from where they can be obtained, through the setting-up of Camden Libraries' Free Leaflets Information Service (FLIS). For an annual fee (E120 in 1992) subscribers receive a copy of the Frills directory of free information leaflet suppliers, a monthly batch of new and updated leaflets and amendments to the Frills directory, and details of a model system for organizing leaflets. Details of this service can be obtained from Community Information Services, Camden Leisure Services Department, St. Pancras Library, 100 Euston Road, London NWI 2AJ. A similar, but not so extensive, service is provided by the NACAB as part of its Information Service, but you would need to have approval to subscribe to the full pack.

Organizing hard information

We have already seen that hard information can be found in a great variety of forms. For the benefit of organizing it, the following categories can be identified:

Material in book form for use by clientele or information workers will be best arranged on shelves in the broad subject groups identified when acquiring the material or according to a classification scheme.

Ephemeral material for use by information workers is best stored in a vertical file using the same subject headings as those in the subject sequence of the information file, though probably with more subdivisions, or using a classification scheme. Alternatively, the material could be kept in file boxes on shelves using broad subject headings.

Ephemeral material for use by clientele on the premises is best kept in file boxes, ring or display binders and ideally interfiled with books. Vertical files are not recommended for public use as they tend to be a deterrent. Another way of dealing with ephemeral materials is to gather them together in packs on particular topics and, for maximum impact, to display them face outwards on sloping shelves.

Ephemeral material for the public to take away can be displayed in special leaflet dispensers, on a sloping surface or on tables. Ideally, they should be displayed in broad subject groups or by originating Organisation, e.g. Benefits Agency, Department for Employment, but this is rarely possible because even within Organisations leaflets are produced in varying sizes.

EXTRA INFORMATION

Many Organisations produce printed material in the course of their work or activities. The most common types of material are directories or address lists of members (individuals or constituent groups), newsletters, events sheets or cards, annual reports, constitutions, posters, leaflets, funding appeals, campaigning literature, advertising brochures, balance sheets and commemorative booklets (anniversaries, etc.). All this material is potentially useful as back-up information to the database but there is no sure way of obtaining it. When sending out the original questionnaire to Organisations, you can ask to be put on their mailing list for this type of material, if one is maintained.

Some Organisations that operate on a shoe-string budget may make a small charge for their mailing list facility. Even when you are put on a mailing list, regular receipt of material may depend very much on the enthusiasm or efficiency of the secretary of an Organisation. Often the best solution to obtaining material is to maintain regular contact with Organisations and to pester them continually. When material arrives, it should be dated, so that you know roughly how old it is, and carefully filed for future use. The simplest method is to have an envelope or folder for each Organisation. Write on the outside of the envelope the name of the Organisation as used in your database and arrange envelopes alphabetically in boxes or a vertical file. You should also add to this file any cuttings from newspapers or magazines concerning those Organisations. The file should be checked regularly, at least once a year, to remove out-of-date material. When writing to Organisations for updated information for your database, ask them for any new literature.

Before throwing away the withdrawn material, check to see if it would be of use to some other Organisation or section of your service which collects material on local history.

THE CLASSIFICATION

It is not my intention to go into any great detail about the classification of community information materials. For most small information services, it will be quite adequate to arrange material by broad subject groups subdivided according to the form in which the clientele's needs and problems are presented.

For ease of labelling vertical files, book spines and boxes, a simple notation can be used based on the initial letter or letters of subjects. For example, Cambridgeshire Libraries' community information service covers 21 categories:

Ben	Benefits	Gov	Government
Bus	Business	Hea	Health
Car	Careers	Hou	Housing
Corn	Community	Law	Legal rights
Con	Consumer	Mon	Money
CR	Community relations	SC	Senior citizens
Dis	Disability	Tra	Transport
Edu	Education	Ump	Unemployment
Emp	Employment	Worn	Women
Env	Environment	You	Youth
Fam	Family		

Some libraries also include as separate categories: Death, Equal opportunities, Fuel (or Energy), Gay rights, Leisure, and Trade union rights. These categories can then be broken down into smaller subcategories as necessary. Taking the earlier example of Housing, a possible range of subcategories with a mnemonic notation might be:

A	Agricultural accommodation	Re	Rented accommodation
H	Homelessness	ReC	Council housing
I	Institutional accommodation	ReH	Housing Association housing
M	Mobile homes	ReP	Private rented
O	Owner-occupied	S	Squatting

Numerals could be introduced for more detailed subdivisions, e.g. O1-Buying and selling a house; O2-Home insurance; O3-Planning applications; O4-Development; etc.

An effective method that can be used where you have only a small number of broad categories (no more than ten and preferably less) is colour coding. Allocate a colour to each subject and attach an appropriate coloured sticker or tape to the spine of each book or file box. Colour coding can be used in conjunction with notation to provide more detailed subdivision of a subject. When a community information service grows to a size where broad categories are not adequate to 'contain all the material and allow efficient retrieval, then it may be necessary to consider using a classification scheme.

Two choices are open to you, either using or adapting a ready-made scheme or constructing your own. There are a number of general classification schemes used in libraries, of which the most well known is the Dewey Decimal Classification. These schemes have been constructed to organize the whole of knowledge and subjects are usually arranged on philosophical or logical principles. Consequently, they may not be sufficiently detailed or treat subjects in the way your clientele express their information needs. Most schemes have a degree of flexibility and it might be possible to adapt them. The NACAB has its own scheme for arranging materials in the broad area of community information, but it hardly merits the description of a classification scheme. However, it is sufficiently detailed and client oriented to be worthy of consideration.

THE EXTENT

In identifying the need for your community information service, you should have reached a decision already on the community to be served. This may be either a geographical community or a community of interest or both. Either way, you will still need to make a decision about the extent of your information base. For example, a neighbourhood information service must decide what information to collect on the wider area-town, county, region, nation-outside its local community. An information service aimed at a fairly wide area may need to limit its interests to a particular sector of the community or to a particular range of subjects, e.g. the low paid, those receiving income support, or those with debt problems. Even an information service in support of a local campaign may benefit from collecting information on similar campaigns or problems country-wide.

DIFFERENT KINDS

The information base of your service is most likely to include the following types of information:

Soft information: *d*details of clubs, societies, Organisations and services; individuals; events, etc. Usually this information will not be available in a published form, or at least not in sufficient detail, and it will be necessary to make a conscious effort to collect it. This information may comprise the major part of the resources file.

Hard information: factual information on a specific subject e.g. benefit rates for single-parent families, how to get legal aid, how to change your name. This information will be available in a variety of Banns, some ephemeral such as leaflets, pamphlets, booklets, broadsheets, posters, periodical articles, etc. Such items are usually kept in a vertical file, in storage boxes or similar receptacles and accessed via some form of index or classification scheme. Other hard information may be in the form of books, multi-volumed reference works, law books, or even audiovisual forms. If there are many items of this type, they will need to be classified and shelved.

Supplementary information: information already produced in a particular format-directories, handbooks, diaries, annual reports, constitutions, newsletters-of Organisations appearing in your information file.

Two

THE FUNDAMENTALS

While mankind has developed myriad ways of applying and controlling power to dominate and shape our environment, through the use of — tools, weapons, machines, fuels, vehicles, instruments, clothing, buildings and roads, metals, plastics and drugs, agriculture, and electricity — the handling of information has lagged considerably, perhaps because the human brain is itself so remarkably powerful.

Until recently, there have been only three major developments in this area: the invention of written (or painted or carved) language, some five or six thousand years ago; that of simple arithmetic operations, using what would now be called a digital representation of numbers, about a thousand years later; and that of printing, about five hundred years ago. With written language, we get the capacity to make a permanent record of information and also to convey messages across space and time: storage, retrieval, and communication.

With digital arithmetic, we get the ability to perform accurate, repeatable manipulations of quantitative data. With printing, we can make many identical copies of the same record and so broadcast a single message to a wide and continuing audience.

Beyond these outstanding advances, until the last hundred years or so, the only progress has been in the engineering department with increasingly plentiful production of more powerful and reliable and efficient, faster and cheaper devices to implement these concepts. In the last hundred years, we have seen the rapidly accelerating advent of a technology so powerful, novel, widespread, and influential that we may indeed call it the Second Industrial Revolution. Its basis is electromagnetic, in many interconnected forms: photography, photocopying, cinematography, and holography; telegraphy, telephony, radio communication, radar, sonar, and telemetry; sound and video recording and reproduction; vacuum tubes, transistors, printed circuits, masers, lasers, fibre optics, and (in rapid succession) integrated circuits (IC), large-scale integration (LSI), and very large-scale integration (VLSI) of circuitry on a tiny semi-conducting 'chip'; and, finally, the bewildering variety of electronic digital computers.

THE SOFTWARE

The programmes, or software, available for computers fall into two classes, *system software* and *application software*. The system software comprises those programmes that are considered indispensable to the general operation of a given computer system, forming what is often termed the operating system of the computer, and generally are supplied by the manufacturer, though alternative operating systems are sometimes available from software houses. The applications software includes all those programmes that are needed by one computer user but not by another, even if such programmes are widely required, and while some manufacturers will sell applications software (as optional additions to their systems), more often this is obtained from software specialists.

We begin with the *operating system.* This may be divided into a kernel of absolutely indispensable programmes and a shell of almost indispensable, so-called utility programmes. The kernel (or nucleus) is also sometimes termed the monitor. To understand why this kernel programme is needed, we consider what happens when we type a character (say an "A") on our keyboard, as part of a message to the computer. Because our timescale recognizes perhaps 1/20 second, while the computer may deal in units of 1/1,000,000 second, if we want to type an "A" we may find ourselves actually transmitting some 50,000 A's to the computer, before our finger leaves the key! If the keyboard mechanism were to transmit the A for only one microsecond, on the other hand, the computer might will be busy elsewhere and miss it altogether. Therefore, it is necessary to establish a protocol (in the usual diplomatic sense, a formal structure for the orderly passage of information) or "hand-shaking" procedure.

This is often done by sending the A on one line (or lines) and simultaneously sending a pulse along an auxiliary line, which sets a "flag" bit in the CPU. When the computer is ready to receive a new character, it checks the appropriate flag bit over and over, until it detects that it has been set. Then it reads the character, resets the flag to its "null" state, and sends a signal pulse back to the keyboard (or its controlling circuitry), telling it that the character has been received. The flag is always open, so the transmit pulse need only be sent once by the keyboard; but the character continues to be sent until it is acknowledged by the computer. Thus a keyboard character will neither be missed nor read repeatedly. In addition, either the CPU or the keyboard controller should send the "A" to whatever device displays the typed characters.

If the particular keyboard were to be the only input device ever to be connected to the CPU, it would be best to incorporate all this in the circuitry of the CPU; but since there may be several different inputs to the CPU, and several types of terminals may be used, coming from different manufacturers, it is found preferable to have the I/O procedures made part of the operating system. In addition to the keyboard input already described, there will be short machine language programmes for input from any other devices attached to the computer and corresponding programmes for output to display screens, printers, and so on. The orderly operation of a computer is subject to interruptions (or *interrupts,* as they are known in Computer jargon), either because a programme contains an instruction that at some point, is not executable (such as a division by zero or a reference to a nonexistent memory location), or because someone needs to terminate execution for some overriding reason (such as the decision that a programme is in error and is generating garbage or is in an infinite loop).

Handling errors and interrupts is another function of the kernel of the operating system, as is the start-up (or "bootstrap") procedure which initializes operation. If, as is often the case, several users are connected to the computer, then the operating system must handle the tasks of Job-scheduling (and Job accounting and billing, if this is appropriate), and the allocation of storage, in main and extended memory, and of other resources (such as printers or communication lines), and the management of *time-sharing* (as between several terminals). Another task that is handled by the operating system is the management of *user files,* and their transfer between main memory and extended memory (such as disk). Indeed, one sees frequent reference to *disk operating systems,* such is the importance of this function of the operating system.

Another function of the kernel is to provide protection and security to users and to itself, both from authorized users encroaching on forbidden territory and from unauthorized users attempting to use the computer.

Ideally, each user should, on giving the correct access code (such as a "password")—this feature did not exist on the first computers, and is still absent in personal microcomputers with only one user/owner-who has access strictly limited to the parts of the computer allocated to him or her by the operating system, but in such a way that the impression is maintained that he or she is the only user present.

Another very congenial aspect of this concept is that some operating systems conduct their memory management function in such a way that the user need make no distinction between main and extended memory; this is referred to as a virtual memory system. The kernel programme has to be able to display, move, modify, and search, at least the main memory, and to initiate execution at any given address, or activate any of the peripheral devices. Using these functions, by means of appropriate commands in the "operating system language," together with its file-handling capabilities, the system can load, compile, or execute any programme stored in its memory (main or extended) written in machine, assembly, or higher-level language, provided that a suitable compiler or other translator is available to it.

We now turn to the utilities provided by most operating systems. First, we have an assembler (and perhaps also a *disassembler,* which translates a programme written in machine language into the more intelligible assembly language; as well as a *macroassembler,* which allows the user to define his own macroinstructions in machine or assembly language), together with a selection of translation programmes, either interpreters or compilers, for the higher-level languages that the user wishes to employ.

The smaller microcomputers will provide an interpreter for some dialect of BASIC, since this a relatively simple language to learn and to interpret into machine or assembly language; beyond this, one must pay for additional languages. There may also be a variety of TRACING and DEBUGGING utilities, according to the cost, size, and sophistication of the computer and its operating system.

There will be facilities for linking or chaining programmes together. While the kernel will contain the rudiments of a file-handling system (to create, destroy, list, locate, and transfer files), there will also be utilities for further management of these. Such programmes will sort, find (among other files), search (in a given file, for information specified), transform, edit, and combine files. Indeed, a good programme editor can enormously facilitate the rapid and painless creation and modification of files, which include both text and programmes.

The line separating the kernel from the shell of utilities is purely conceptual and far from sharp. Similarly, the boundary of the entire operating system is far from definite. What is available beyond the essentials mentioned above will be called part of any decent system by some, part of a compiled language by others, and just applications software by still others, depending on their point of view. The ability to handle a variety of data structures (such as arrays, lists, strings of characters, trees, queues, and stacks) may come from the use of a suitable higher-level language or from an extended "system development" utility package.

Similarly, languages intended for scientific and engineering applications usually handle floating-point and multiple-precision arithmetic, though this may be provided even in computer hardware. The same applies to routines for computing, for example, sines, cosines, logarithms, etc., and pseudorandom numbers (the last beloved of, computer games inventors!).

A *database management system* (DBMS) may be just a glorified file-handling utility; or may be an elaborate programme, cross-indexed and relational, with its own language of special commands, for answering any conceivable question about a large amount of intricately structured data.

As is often the case, you get what you pay for, both in money and in memory space; and vendors' claims have to be carefully scrutinized and verified. It is advisable not to overbuy one's capabilities beyond one's needs.

Similarly, a graphics package may simply allow one to produce passable graphs, histograms (i.e. bar-graphs), pie-charts, and perhaps games, with a resolution of some 200 by 300 delightfully named pixels; or it may allow you to resolve perhaps 2000 by 3000 pixels, in a variety of colours, with the ability to draw complex three-dimensional shapes, properly shaded, illuminated, and textured, and move them by commands in a special language. The latter systems are a lot of fun to operate, and can be most helpful to draughtsmen, animators, film designers, and simulator-trainer designers; but they cost a bundle, and they require quite powerful computers to hold and run them.

Another offshoot of the file handier is a cluster of programmes for generating reports, journal ledgers, and accounts of all kinds, as well as forms and mailing' lists (the latter leading to a kind of DBMS in which mailing lists are matched to interests and characteristics of the individuals itemized).

Again, there are statistical packages of various degrees of sophistication, from a mean-variance-covariance calculator, to systems able to analyse very complex sets of data by elaborate techniques, using a whole statistical computer language. What has been described as "the most popular programme ever written" is usually given the generic name of an *electronic worksheet,* though the trade name of the first such programme, Visi-Calc, like Kleenex and Vaseline, has become almost generic.

A table is presented on the video screen and each entry is allocated either a numerical value or a formula relating it to other entries. When the data are sufficient, the resulting numbers are displayed. When an entry is changed, it and all entries depending on it are altered accordingly at once. This provides a representation of a given situation surpassed only by a graph in its impact, and a first-class planning aid. Finally, among the borderline system/applications software utilities, will put what is usually called a word-processing package (though would prefer either character or text processor). This is an extension of an editing utility, in which the text may by "scrolled" up and down on the screen, edited in the usual ways, but also more specifically for producing letters, articles, reports, and other written copy.

Margins may be set and the text right, left, or double justified, or set up in multi-column pages, with page numbering, indexing, and even the use of different sizes and styles of typeface, in the most sophisticated systems. Here again, cost rises steeply, and one should buy only what one will need.

Beyond these programmes, there lies an endless variety of unquestionable applications programmes. There are programmes for ballistics, boat design, analysis of molecular structure from x-ray diffraction data, tabulation of Bessel functions, simulation of naval battles and economic cycles, etc.

Most of these are not in the market and only work on one machine; but there are very many programmes available in the open market, produced or distributed by software houses, with various levels of efficiency, and sophistication. *Programming* is a fascinating, intricate, rewarding, but unforgiving and at times infuriating occupation. We will encourage all of you who have the time to try it. If you have a computer at your disposal, it would be a shame not to learn a simple language, such as BASIC or PASCAL and try your hand at writing a simple programme or two. The sense of achievement when you have a working programme is great; perhaps because the process is addictive and consumes much more time than you would believe possible.

TECHNICAL LANGUAGE

The CPU of any computer is designed to accept and execute a specific set of *operation codes (op-codes),* ranging in number from a dozen or so to several hundred. Different makes and models of computers and microprocessors may have entirely dissimilar op-codes; but the operations that they represent are much more alike than different, both through functional necessity and historical development.

The interpretation of the op-codes is built into the hardware of the OCU (though sometimes the details of interpretation may be modified by the user through what is called *microcoding),* and it is part of this interpretation that the complete instruction being decoded contains a certain amount of further information (such as parameters, indices, and one or more memory addresses).

The "length" of a computer instruction (the number of consecutive memory words, or of bytes, occupied by it) may vary, but the OCU automatically adjusts to this. An executable programme (sometimes called "object code") consists of a sequence of machine instructions, consecutively stored in the computer's memory and (with the exception of "jump" instructions) executed in the order in which they are stored.

The aggregate of possible machine instructions is called the machine language. In the computer, a programme consists of a long string of binary digits (bits), usually written as 0's and 1's; and, of course, the same programme would be interpreted quite differently (usually as nonsense) by any computer for which it is not designed. Indeed, the slightest error in a programme almost always leads to an error in its output (usually a fatal error!). This state of affairs is sometimes expressed by the computer adage, "Garbage in; garbage out," or just "GIGO." Errors in programmes are called bugs; and the tedious, odious process of finding and correcting such errors is termed debugging. It is estimated that, in the production of a working programme, the debugging time may be two to four times as long as the time it takes to plan and write the programme initially.

To give the reader a feeling for the nature of machine language, we present a simplified, fictitious, but typical, machine language specification. Our computer has two 16-bit accumulator registers (acc), X and Y, which may be coupled into a single 32-bit ace XY, with X holding the more and Y the less significant digits; these are attached to the ALU; and a programme control register (pc) Z, also of 16 bits, attached to the OCU, which contains the address of the next instruction to be executed. Instructions will contain a 2-bit acc code a, with a = 0 referring to X, a = 1 to Y, a = 2 to XY, and a = 3 to Z. The memory consists of 216 = 65536 16-bit words, directly addressable with a 16-bit address, n.

There are four addressing modes, denoted by a 2-bit mode code d, with d = 0 referring to absolute address (address *n*\0 refers to the actual number *"n"), d* = I to direct address (address *n*\1 refers to word n in the memory), *d = 2* to indirect address *(n*\2 refers to the memory address which is the content of the memory word with address *n),* and *d* = 3 to relative address (n\3 refers to the memory address that is the sum of *n* and the content of the pc register Z, with any carry to the seventeenth bit ignored). We will write {{*a*}} for the acc denoted by a, *(n\d)* for the memory word with address *n* in mode *d,* and C[*x*] for the content of word or acc *x* (so, e.g., C[{n]2}]=C[C[n]], while C[{n\3}]=C(n+C[Z]). *A port code p* selects one of 8 *output ports (p=0,* 1, . . ., 7) and 8 *input ports (p = 8,* 9, . . . , 15), these ports transmitting 16 bits at once and having to be reactivated before each function (i.e., each output instruction sends out one 16-bit number, and the machine must wait for the output to be acknowledged before making another output; while each input instruction reads one 16-bit number, if an input is available, and the machine must wait for a new input each time). Finally, a bit code b refers to each of the 16 bits in a word (bit 0 being the least significant-rightmost and bit 15 the most significant-leftmost). In some cases, the codes a, d, p, and b are interpreted somewhat differently, depending on the particular op-code c.

As an example of a very simple computer programme, we consider the solution to the following computer problem. Our computer is to be fed a sequence of one thousand 16-bit numbers at input port #9 (these may be keyed in by hand or fed in by a digitizer connected to some experiment). They are to be stored in memory words with addresses 5000, 5001, 5002, . . . , 5999. Their sum is to be computed and stored in address 6000 and output to a printer through output port #2. The programme is to be stored beginning at address 0 in the memory.

Again, what is important about this example is not its detailed form, but the difficulty of its interpretation, and therefore also the difficulty of verification and debugging.

The programmer must deal with a mass of details that are of a purely mechanical nature and have no relevance to the problem being solved. "Higher-level languages" are attempts at making the computer understand the programmer's way of thinking, rather than forcing the programmer to think like a machine.

TECHNOLOGICAL CHANGE

The progress has been truly amazing. In only about 40 years, electronic communications and news media have become commonplace and indispensable. Computers have proliferated, becoming increasingly fast, powerful, small, and cheap, so that now there is scarcely a human activity in which they are not to be found, bearing an increasing share of the burden of repetitive information processing, just as the machines of the First Industrial Revolution have taken over the majority of heavy and unpleasant physical labour. Now, information can not only be stored, retrieved, communicated, and broadcast in enormous quantities and at phenomenal speeds; but it can also be rearranged, selected, marshalled, and transformed.

Until recently, these activities were the sole domain of the human brain. While creative, judicious, moral, and aesthetic choices are still best left to people, all the tedious and mechanical mental processes can now be relegated to the accurate, fast, and tireless machines. Any sequence of operations on information that can be precisely specified can be carried out without further human intervention or supervision.

At first, computers were the experimental toys of university researchers; then they became the tools of government establishments and giant corporations, huge, expensive, individually designed and manufactured, and beyond the reach of any but the wealthiest Organisations.

People thought of the future in terms of machines of ever-greater speed and capacity; centralized behemoths would hold all the world's information in gigantic data banks, whence major decisions would issue to be imposed upon the populations at their mercy.

With the emergence of powerful, cheap, mass-produced computers-on-a-chip, the picture has changed radically. Now we see tiny computers everywhere: in wrist-watches, microwave ovens, electronic games, pocket calculators, cameras, typewriters, musical instruments, etc.

What used to be done, with few options, by intricate mechanical devices is now performed, with great flexibility and convenience and at much less expense, by the ubiquitous pre-programmed microcomputer.

The probable future has become one of millions of small yet powerful computers, controlling virtually every machine and appliance. These are distributed in every home, on every desk, in every workshop; many of them connected in a maze of small and large networks, much like the present telephone network. This enables individual computers to communicate, sharing information in a gigantic distributed database, and gaining, through distributed processing, computational power whose extent is yet difficult to gauge; all this following the individual requirements and choices of the owner or operator of each machine. Increasingly, we are confronted, not only with the results of the use of computers throughout industry, commerce, banking, advertising, science, the communications industry, newspapers, airlines, and hospitals; but with the realistic possibility of purchasing computer power for our own small enterprises, offices, and homes. This may be done in a variety of ways; but in all of them, the real cost of computation is constantly diminishing.

It is probably fair to say that the question of computerization is not "whether," but "when" and "how." We must choose whether to lease equipment or to buy it; whether to install terminals, connected to a computerized "service bureau," or a complete local computer system; whether to get a package of standard programmes directed towards our kind of work, to commission new programmes tailored to our special needs, or to learn programming and write our own; whether to go it alone or to share a system with a number of similar users (especially if they are in the same building); how far to take our first efforts at computerization; what to include and what to continue doing by hand. Computer programming is likely to become the literacy medium of the third millennium AD.

Elementary schools may well be teaching it before long, and we might be well advised to gain at least a smattering of knowledge of computers and of programming competence, especially since computer languages and programming environments are becoming increasingly helpful and friendly to the uninitiated user.

A computer is a machine for the automatic processing of information. Historically, this information was numerical, and computers were machines for doing arithmetic.

Unlike the simpler calculating machines, which can perform only one elementary arithmetic operation at a time, they need to be told what to do next (usually by suitable button-pushes); computers can be given a list of operations to perform (often with branching and repetitions, depending on tests of sign or value included among the operations), which they will then execute in proper sequence without further intervention. This sequence of instructions is called a *programme*.

A digital computer stores its information in the form of words, finite ordered sets of digits, each of which can have only one of a finite set of values.

Considerations of simplicity, reliability and economy dictate that electrical engineers should design computers to consist of a great number of similar pieces of circuitry, each of which can only be in one of two states, usually denoted by 0 and 1.

Such *binary digits* (or bits) are the elements of which computer digital representation is built. A row of eight bits is called a *byte,* and the majority of computers have their storage of information organized in words of one, two, four, or eight bytes (8, 16, 32, or 64 bits).

The number of bits in a word is termed its *length;* if this is *k,* then the number of possible distinct pieces of information that can be stored in such a word is 2. In particular, four bits together can have 16 different contents, and these are standard binary representations of the numbers 0-15:

0000 = 0 0001 = 1 0010 = 2 0011 = 3

0100 = 4 0101 = 5 0110 = 6 0111 = 7

1000 = 8 1001 = 9 1010 = A 1011 = B

1100 = C 1101 = D 1110 E 1111 = F

with A = 10, B = 11, C = 12, D = 13, E = 14, and F = 15. These may now be viewed as the 16 possible digits of a representation (the hexadecimal, or hex), which is much more compact and humanly intelligible than a long string of zeroes and ones. For example, the byte 10110010 becomes "132," and the four-byte computer word 01001100011100101101000110001110 becomes the eight-digit hex word "4C72D18E."

The following are various parts of which any computer is composed. These are, essentially,

1. A *central processing unit* (CPU), which is the controlling and computing centre of the machine;
2. A *memory,* possibly of different levels, in which both *data* and *instructions* are stored;
3. A variety of *input* and *output (I/O) devices,* through which the machine communicates with the world outside it.

The CPU consists of an operation control unit (OCU), an arithmetic/logical unit (ALU), and a relatively small, very-fast accessible local memory (LM). The OCU keeps track of the memory location of the next instruction to be executed, and analyses the current instruction, so as to activate the proper operation of a memory transfer, non-sequential jump (by appropriately changing the address of the next instruction), input or output of information, or computation (performed by the ALU), as is indicated by the instruction code.

The ALU actually carries out the elementary arithmetic operations (addition, subtraction or negation, multiplication, division or reciprocation) and logical operations (these being bit-by-bit operations, without carry' such as not, and, xor; e.g.., not 1010= 0010, 1100 and 0101 =0100, 0110 xor 1010 = 1100) on the data given to it by the OCU. The LM receives the operands called for by the OCU and also the results of the ALU'S operations upon them. For example, the OCU may retrieve the factors of a multiplication from the main memory into a pair Of LM registers and instruct the ALU to multiply them and place their product in another LM register. Such registers are usually called accumulators, and they are normally double-length (since the product of two k-bit numbers is a 2k-bit number). Other LM registers are used for counting (e.g. repetitions) and are called index registers; others hold intermediate values and are called buffers; and, finally, there are one-bit or two-bit registers which reflect the information on which tests are made by the OCU (for example, the occurrence of a carry, the vanishing or positivity of an answer, or the occurrence of an arithmetic overflow): these are termed flags.

Originally, the CPU was a sizable piece of electronics, hand-assembled and highly complex. With the advent of micro-miniaturization of circuitry, printing, and photographic techniques, and the mass production of components only the largest computers (mainframes) are built in the old way. Smaller systems generally have the entire CPU on a single chip. Among these, the name microcomputer is now applied to those with less than a million words of memory and a word length of one or two bytes; the name minicomputer applies to the larger machines, with two to four-byte words and one to a hundred million words of memory. (The smallest micro is probably more powerful than the big computers used by universities and industry in the 1950s.)

The *main memory* (mm) consists of magnetic or electronic components which store the information (both data and instructions) needed by the computer. The individual words are directly addressable from the CPU by number (rather like houses in a street), and their contents are retrievable in very short times, of the order of the operation time of the CPU (ranging from fractions of a *nanosecond,* 10^{-9} or one billionth of a second, for the fastest main-frames to several *microseconds,* 10^{-6} or millionths of a second, for the slower micros). This is often referred to as high-speed storage or random-access memory (RAM). While most of the mm is erasable and may be changed at will, some memory is used to store constants and often-used utility programmes and is not erasable by the CPU: such memory is called *read-only memory* (ROM). Sometimes this is optional and can be plugged into the computer: this is called *firmware.*

Note that a computer with one-byte-long addresses can have at most 28= 256 words of mm; a two-byte address can reach 256^2 = 65536 words; a three-byte address can select any of 16777216 words of mm; and so on. In practice, most micros and minis have mm ranging in size from 2 14= 16384 to 218= 262144 words. It should be noted that 10^3= 1000 and 2^{10}= 1024. Because computers are so heavily slanted towards base-2 representation, it has become almost universal computer parlance to use the prefix *kilo* or K (which usually denotes a thousand units) to denote 1024 and the prefix *mega* or M (which usually denotes a million units) to denote $2^{20} = 1024^2 = 1048576$. Thus, we write 16K for 2^U and V4M for 2^{18}. Perhaps the commonest size of MM consists of 64K = 65536 words.

Almost all computer instructions comprise an operation code (usually one byte long, allowing 256 possible operations to be specified), followed by an operand reference (number, index, or address) of variable length (since some operations require more data than others; for instance, the STOP instruction needs no operand, so it is one byte long).

The *extended memory* (EM) is sometimes considered under I/O devices, both because it is often physically located outside the computer (while the CPU and the mm are usually in the same enclosure), and because its speed of access is much slower than the speed of operation of the CPU and is comparable with the range of speeds of I/O devices. Most read/write memory is magnetic (optical memory is read-only, and fast RAM is either magnetic or electronic), either in the form of tape, drum, or disk, coated with magnetic material, much like sound recording equipment; it is similarly erased, read, and recorded upon by "heads" which contain electromagnetic sensors/ polarizers. The cheapest (and most limited, in both speed and capacity) device is a common portable cassette recorder, with 1/4" tape cassettes. From this, one moves to specially engineered cassettes and recorders, and thence to high-speed 1/2" or I" reel-to-reel drives carrying thousands of feet of tape at very high speeds. Access times can be quite good for sequential access, along the tape, but random access time is poor at best, running to seconds or even minutes. Economy and a virtually unlimited total storage capacity (on numerous cassettes or reels; but only as many units as one has on-line tape-drives are actually accessible without human intervention) are the only advantages.

NEXT STEP

After some introductory description of an imaginary computer, and especially of its CPU, and the establishment of some essential notation. Presented with a set of arithmetic and other transformations required by potential users of a proposed new computer, the electronic and logical design engineers seek the simplest circuitry that will execute operations sufficient to generate all the required transformations. Circuits must be simple to be fast, efficient, reliable, and cheap. However, when an instruction has 16 or more bits, most of which should for the sake of efficiency, have some significant effect; the exact and complete specification or the action induced by it may well be somewhat forbiddingly intricate! And indeed, simplicity of circuitry does not usually lead to simplicity of use.

An ample of a programme in this machine language, which takes 16 instructions and 28 16-bit words of memory storage, to read in one thousand numbers, store them in consecutive memory locations, sum them, and print out the sum. However, please note that the English description of what the programme does (though much easier to say and comprehend) itself requires 103 characters, each equivalent to 8 bits; so that it is almost twice as long as the machine-language programme. Indeed, the difficulty of the latter lies not in its length, but in its alien and opaque form of expression of what it does. The first improvement on machine language was assembly language. This is not very different from machine language, but removes the most glaring and trivial irritations. First, instructions may be labelled, so that one no longer needs to count lines to address jump instructions; one jumps to the label. This is also a great help when one wishes to insert additional instructions, either to correct or modify a programme (every jump need not be changed). Second, memory may be labelled, so that numbers may be stored in symbolic addresses and retrieved therefrom, leaving it to a computer programme to assign actual storage. Third, the entire programme becomes relocatable anywhere in the computer memory. Finally, and perhaps more trivially, the op-codes may be replaced by abbreviated mnemonics, and the instruction layout may be relaxed, using punctuation marks. Of course, a programme written in assembly language is no longer directly intelligible to the computer or executable by it. It becomes a piece of textual input, to a "translator programme" called the assembler. Naturally, the programme may be stored in the computer's memory, like any other piece of text; but, before it can be executed, it must be "assembled." The assembler finds appropriate space for the object code (i.e. the machine language equivalent of the input programme) and for its needed storage space (often called variables). The assembly language instructions correspond, one-to-one, to machine-language instructions.

For example, we may denote memory locations by one, two, or three lower-case letters (both for labelling instructions and naming variables). A numerical address will still be allowed. Absolute addresses (i.e. actual numbers; d=O) will correspond to quoted numbers (e.g. "6000"); direct addresses (d=l) to variables named as stated; indirect addresses (d=2) to variable names placed in parentheses; and relative addresses will not be used. With relocatable programmes and symbolic addresses, they are no longer useful). Accumulators (X, Y, XY, or Z) will not be referred to symbolically, but only by these names. Instruction labels will precede the instruction and will be terminated by a colon (:). Spaces will be ignored. Op-codes will be denoted by strings of capital letters and terminated by a comma (,), which will also separate multiple arguments.

In the assembly language notation, A denotes X (a=0), Y (a= 1), XY (a=2), or Z (a=3). Italic letters (p, x, or b) denote numbers (possibly in quotes, to denote absolutes) or variables names (possibly in parentheses, to denote indirect addressing), and are interpreted according to the op-code (e.g. A cannot be XY in IN and OUT instructions; if A is XY in instructions with c = 1, 2, and 4 (with p > 5), the operand must be 32 bits long-either a double-length number, or an address referring to the first of two consecutive 16-bit memory words. Again, the reader must remember that it is not the particular details of our specifications that are important or memorable but the kind and degree of detail occurring.

Assembly language mnemonic	***Machine language instruction***
IN A, (p-8)	$c = 0, a$ ** 2, $p > 8$
OUT A, *p*	$c = 0, a$ ** 2, p < 7
TO A, x	*c*= 1
FROM A, x	$c = 2, d > 1$
FROMA, B	$c = 2, d = 0$ -
CL, A	$c = 3, d = 0$
NEG A	$c = 3, d = 1$
NOT A	*c* = 3, d = 2
RNDx	c = 3, *d* = 3, a = 2
ADD A, x	c = 4, p = 0
SUB A, x	c = 4, p = 1
AND A, x	*c* = 4, p = 2
OR A, x	c = 4, p = 3
XOR A, x	c = 4, *p* = 4
LADD, x	c = 4, p = 5
LSUB, x	c = 4, p = 6
LAND, x	c = 4, p = 7
LOR, x	c = 4, p = 8
LXOR, x	c = 4, p = 9
CSHL A, P	c = 5, *d* = 0
CSHR A, P	c = 5, *d* = 1
ASHL A, P	$c = 5, d = 2$
ASHR A, P	$c = 5, d = 3$
JMP, x	c = 6, *p* = 0

contd.....

Assembly language mnemonic	***Machine language instruction***
JPCZ A, x	c = 6, p = 1
JPCNN A, x	c = 6, p = 2
JPCNZ A, x	*c* = 6, p = 3
JPCP A, x	c = 6, p = 4
JPCB A, 0, *b,* x	c = 6, p = 5
BIT A, 0, *b*	c = *6, p* = 6
BIT A, 1, *b*	c = *7, d* = 1
BITNOT A, *b*	*c* = 7, d = 2
STOP	c = 8
INTER	c = 9

Figure 1: Typical assembly language mnemonics

Consider Figure 2. Here, numbers are given in decimal notation. Note the tremendous improvement in direct legibility of the programme in comparison with the binary strings. The assembler is left to figure out where to store the sum (the variable "sum" which the machine language programme put into address 6000) and the current address of the listed numbers (the variable "w," which the machine language programme puts at location 28 if the programme begins at location 0, necessitating counting the words occupied by the programme; if the programme were to be modified or relocated, this would have to be changed, and would be an unnecessary source of programming errors); and similarly, there is no need to count words to determine the addresses of "ret" and "out" (placed by the machine language programme at 5 and 24 ' as it happens).

As an exercise, the readers may wish to try to modify the programme so as to allow for the possibility that the sum of the 1000 numbers might occupy more than 16 bits. In machine language this requires no less than 12 changes and seven shifts of memory location.

	CL X; FROM X, SUM; TOX, '5000'
ret:	FROM X, W; SUB X '6000'; JPCZX, out
	INX, 1; FROM X, (W); ADD X, sum
	FROM X, sum
	TO X, W; ADD X, T; JP, ret
out:	TO X, sum; OUT X, 2; STOP

Figure 2: The summation programme in assembly language

In assembly language only the substantive changes (from the programmer's point of view) need be made. The amended programme is given in Figure 3. Here we note that only seven changes are needed; but observe that "sum" becomes a double-length address-just another piece of bookkeeping handled automatically by the assembler.

```
     CL XY; FROM XY, SUM; TOX, '5000'
ret: FROM X, W; SUB X '6000'; JPCZX, out IN X, 1; FROM X, (W);
     TO XY. sum ADD XY. (w);
     FROM XY, sum TO X,W; ADD X, T; JMP, ret
out: TO XY. sum; OUT X, 2; OUT Y. 2: STOP
```

(Underlined material represents changes from previous programme). Alternative version would have third line replaced by:

```
     CLX: IN Y 1, FROM Y (W); LADD, sum FROM XY sum
```

Figure 3: The summation programme modified for long sums

Of course, a programme in assembly language still represents, instruction-by-instruction, a programme in machine language, and therefore still has the defects of "walking the road in extremely short steps," as we put it earlier.

All that shuffling of numbers in and out of the accumulators, counting and restricted testing, is not natural to us, and its mechanical nature suggests that an improvement is still possible. Thus it was not long before programmers devised much more humanly natural languages, which are generically called *higher-level* (or *algebraic) languages*. Some of the more common ones are ADA, ALGOL, APL, BASIC, C, COBOL, FORTRAN, LISP, PASCAL, PL/I, RPG-II, SNOBOL, SPITBOL, WATFOR, and WATFIV. There are more; and all come with a variety of versions and dialects.

The characteristic that puts a language into this class is that one higher-level instruction translates into several assembly or machine language instructions. Beyond this, the languages differ according to the kind of programme they are intended for.

While most languages will perform most tasks, FORTRAN was clearly intended to do scientific and engineering computations, COBOL to perform business data-processing, LISP to manipulate lists, and SNOBOL, strings of text. As a final illustration of the power and intelligibility of such languages, we present higher-level versions of our summation programme in BASIC, WATFIV, and PASCAL. Once more, note that the details of the individual languages do not matter to us at this point (though, if the reader ever intends to programme, then some language will have to be thoroughly understood; but assume that many of my readers have no such intention); it is their general flavour and appearance that is noteworthy; and the casual readability of the samples presented is their salient characteristic.

Of all the hundreds of programming languages devised, it is safe to say that more lines of programme have been written in FORTRAN or COBOL than in all others combined, though this reflects their age rather than their desirability. The fast-gaining runner-up must surely be BASIC, which comes with every micro and most minicomputers; though some would say that BASIC, which perpetuates the style of FORTRAN, spoils any chances of producing really good programmes and should be replaced by PASCAL. Fashions come and go in programming languages, not always fully based on rational arguments, and personal preferences vigorously touted.

Given a higher-level language, it is necessary to have a translator programme to turn it into machine (or more frequently assembly) language. Such translators are of two kinds.

An interpreter operates at execution time: beginning with the first instruction of the stored higher-level programme, it translates it into one or more machine language instructions, which it proceeds to execute; then it reads and translates the next higher-level instruction and executes that; and so on. If a jump is encountered in the source programme (as the higher-level language programme is often called), the interpreter goes to the next instruction in accordance with the jump and proceeds to read, translate, and execute it, without taking into account whether it has already encountered it. Thus, the same source instructions will have to be passed and translated a thousand times (a very time-consuming and wasteful procedure). But the (usually much longer) object programme need not be stored, and interpreters themselves tend to be shorter and simpler programmes. This is why a BASIC interpreter is a natural adjunct to a microcomputer, which tends to be a little cramped for space; especially since many microcomputer users are riot interested in extremely lengthy computations.

By contrast, a *compiler* is a programme that translates a source programme into an object programme, the former being in higher-level language and the latter in machine or assembly language.

The programme is translated as it sits in memory, not during execution; so that the problem of repeated translation is avoided. Indeed, there are many so-called "optimizing compilers" which pass through the compiled object code several times, eliminating redundancies and inefficiencies to produce faster object programmes (something that would be impossible for an interpreter to do). Speed is greatly increased, in general, at the expense of space in the computer memory. Of course, once a programme has been compiled and proven to be free of bugs (debugging is far easier in source code than in object code), the source programme may be stored for future use and only the object programme kept in main memory. Compilers are available for almost all higher-level languages on almost all machines, and are essential for extensive applications.

MEMORY POWER

When we wish for practically useful EM, combining large capacity with relative economy and speed of random access, we must turn to drum or disk memory; and, nowadays, the former have been practically replaced by the latter. Disk memory is of two types: floppy disk and hard disk, the first being the cheaper, slower, smaller-capacity option. Floppy disks are flexible, have diameters of 51/4" or 8", generally, and are removable from the disk-drive, so allowing one to build up an unlimited library of stored data. The information is stored on concentric circular tracks (not on a single spiral track, as on a sound record), on one or both sides of the disk. The number of tracks and the number of bytes per track vary (the density increasing with precision of engineering, and so with cost of the drive), but the total capacity of a floppy disk is in the range of 50KB to 1MB. The disks rotate at, typically, 300 rpm, and access time is governed by the time required to place the movable head on the right track, a fraction of a second, plus the fifth of a second taken by the head to traverse the circumference of the track, in search of a record; thereafter, consecutive bytes are accessed at some thousands per second.

Hard disks are rigid and have larger diameters. There are drives which range from anything from one to a dozen disks, rotating at about ten times the speed of floppy-disk drives (and so diminishing the access time of records in a track) with one or several heads.

Fixed-head drives naturally must have a head for each track (which costs more), but save head-movement time in random access. Winchester disks are movable-head drives with sealed-in disks, where the heads ride very close to the disk, cushioned by the layer of air between. In floppy-disk drives, the head actually rides on the disk, eventually wearing it out.

The capacity of hard-disk drive ranges from 10 MB to 100 MB in a single drive. Some movable-head hard-disk drives have removable disks or disk-packs, allowing for greater library storage.

The I/O devices are the computer's link with the outside world. In large mainframe computers, we see *paper card* (as in Hollerith or "IBM" cards) and paper tape readers and punches:

> The perforations in the paper carry the information. Increasingly in large computers, and almost universally in small ones, the main input is from the keyboard of a terminal. This is much like a typewriter keyboard, and depressing any key sends an 8-bit code to the computer.

When the computer is waiting for input from this terminal, it reads the code and interprets it as a datum; when it is busy, either the message is lost or it is held in a "buffer" for subsequent input (this depends on how the connection is made and what the computer is made to do). It is quite common for the computer to be connected to several terminals, all competing for its attention. This is called time-sharing. The computer cycles around the terminals, look for their several inputs while dividing its CPU time among them. The main output of the computer is to the display devices of the terminals; these are either video displays (cathode ray tubes, CRT, just like the screens of black-and-white or colour TV sets; indeed, simple micros sometimes use ordinary television sets as display devices) or printers (in so-called hard-copy terminals). Of course, the computer may be connected to additional video displays and printers, of different qualities, as well as to plotters, a kind of printer for drawing graphs and diagrams. Many types and speeds of printers exist. Other input devices, such as audio amplifiers (receiving signals from radio tuners, record players, etc.), video receivers and recorders, laser disk drives and a variety of scientific instruments, can all be classified as transducers (devices that transform physical quantities, such as position, conductivity, pressure, temperature, vibration frequency, or amplitude, into electromagnetic impulses) linked to digitizers (which convert such impulses into sequences of zero/one pulses). Output from the computer can similarly follow the reverse process, yielding visible or audible results, or the control of mechanical or electrical equipment. Thus, computers can draw pictures (often, moving pictures), make music and other sounds, and can control appliances, machinery, and whole manufacturing processes.

In order to connect remote terminals to a computer, use is often made of telephone lines; and, for this purpose one employs a device called a modem (for "Modulator/ Demodulator"), which converts the computer's digital signals to and from telephone audio signals. This has a cradle shaped to hold the ear and mouth pieces of a telephone's hand-set. It is also possible to connect several computers in this way. This is called the formation of a computer network. Of course, computers may also be connected by cable, fibre-optics, or microwave link. It should be noted that terminals usually, and other I/O devices often, themselves contain computers of varying degrees of power and complexity; so that even a multiuser computer with a central CPU may still be seen as a kind of computer network in its own right. The computers in I/O devices such as terminals or printers are often referred to as peripheral processors (PP).

Many networks do not have a central computer at all; but are simply a collection of independent computers linked for the sharing of information and, sometimes, computing capabilities. Often, they permit the exchange of messages *(computer mail)* and the pooling of data *(distributed database).* They may also share a common bank of memory, accessible to all.

Finally, since the invention of computer networks, designers have been investigating the possibilities of computers made up of an array of CPUs *(multicomputer, parallel processors, distributed processing).* These new ideas are very powerful and far-reaching: they will probably revolutionize our ideas of computers, and of their applications, in the next few years.

The various peripheral devices are connected to the CPU without examining how. In fact, this may be done in several ways. We can simply have a separate connection (or port) for each device, but this limits rather severely the number of devices that may be connected to the CPU. Another way is to have a single bus (or connection) to which any number of devices may be attached.

The information signal must then carry an appropriate address. The bus receives all signals, and individual devices (including the CPU) seek out and decode only those addressed to them. It is also possible to have a switching device, which receives addressed data and directs them to the appropriate recipient device, rather like a central post office.

The decision on what communication arrangements to adopt is made on the basis of considerations of cost, capacity, and speed. It is often the case that the several devices forming a computer or a computer network have their data coded in different ways.

It is then the job of the CPU(s) and PPs to share the work of interpreting signals into appropriate codes for each machine. This is broadly termed the problem of interfacing devices. Sometimes, the solution is to have a standard code or structure for the communications device (one meets the s-100 bus, the rs-232 serial port, the ASCII character-code, and so on).

Another interfacing problem arises from the difference in the rate at which different devices can send and receive information (this is measured by the baud rate, named after Baudot, the inventor of the first five-hole paper-tape code; one baud is one bit transferred per second; hence kilobaud, kb, and megabaud, Mb typical rates range from 200 baud to 200 kb). One solution is to send each piece of information (usually one character at a time, which takes 8 to 10 bits) only when the last has been acknowledged (this is referred to as a handshake); this is sure, but slow. Another way is to use a storage buffer in which a large batch of information is accumulated for fast transmission, thus not wasting the time of the faster device.

One last kind of choice must be mentioned: some channels of communication are serial (they transmit one bit at a time), while others are parallel (they can transmit a byte or a whole word at a time); the latter are obviously faster, more complex, and more expensive.

When devices are connected by cable, the degree of parallel communication is exhibited in the width of a flat ribbon cable, carrying several wires, side by side, and in the number of pins in the plugs and sockets by which they are connected to the machines. Parallel transmission is a variation on multiplexing. What we have described is generally referred to as the hardware of a computer.

By contrast, the programmes that make the computer work, the "soul" of the machine, as opposed to the hardware "body," are collectively called its software. Inevitably, there came to be programmes that were hard-wired (in the now outdated phrase) into the computer, in the form of ROM. These are termed firmware.

A computer without software is a helpless set-of circuits, and the expertise required to create the basic software that will bring the machine to useful life is comparable to that required to design the machine itself. Indeed, these days, computers are designed in cooperation between computer architects, who design what the computer will do, hardware engineers, who design how it will be constructed to be able to do it, and software engineers, who design and programme the operating system that will run the machine. Beyond this, the computer will also need application software of many kinds, to enable it to do a variety of jobs, such as file-handling, accounting, statistics, payrolls, inventories, complex graphic displays, games, and so on. Typically, the application software is written (i.e. programmed) by the endusers (if they are sophisticated enough) or firms of consultants and programmers and system analysts, often called software houses.

Sometimes, a software house will produce part or all of an alternative operating system, or an addition to an operating system, to make a computer more flexible or more efficient than the manufacturer's own software system allows it to be.

Three

BASIC CONCEPT

Information is one of those misunderstood concepts. Yet it is at the same time one of the most used. We go to the bus station to seek information from timetables. We obtain information from government offices such as the DSS and the DVLC. Banks, supermarkets, leisure centres, libraries and even the police ply us with more. Newspapers, television and the radio present us with their own ideas of what information should be. In many instances, however, we may not agree with their conclusions. McLeod termed its subjectivity as being one person's junk and another's treasure.

We all understand and deal with this accordingly when extracting news from the media, for example. Yet many of our Organisations consider information to be something more. They see it as a vital resource, to be managed like any other valuable resource. How it is used and disseminated through the available technology can determine how efficient, and indeed effective, an Organisation is. Peter Drucker sees information acquired in a systematic and purposeful way as enhancing an Organisation's productivity.

Information is important, we cannot operate without it. But more than this, we are discovering that our ability to process it by increasingly sophisticated technological means is fundamentally changing the way that employees perceive their Organisational environment. The consequences of this could be either to break down established functional controls or indeed to enhance them by becoming super-efficient. Much will depend upon the characteristics of the Organisation before implementation.

As firms become increasingly 'information dependent' we need to identify the perceptual relationship between ourselves and the information we use. How, for instance, can we possibly design an adequate information system if we do not understand the nature of information? The answer to this lies, in part, with the conventional wisdom of the day. Managers, like everybody else, develop their views through exposure to established ideas. Thus, the way in which they understand information will influence the way in which they treat it. Such conventional wisdom is coloured by the technicians, on the one hand, and the user's own experience on the other.

Anecdotal prescriptions abound as guidelines to the way presented for all who care to listen. These arise from a blanket of professional and academic thinking which surrounds the business environment and provides remedies for action. We could categorize all these ideas into two bodies of thought, two paradigms. The first could be termed the resource-driven paradigm. This is because its central theme in understanding information is the continuity and consistency of the information itself. It is very much in vogue at present.

The second body of thought is the perception-driven paradigm. Information is seen as an abstract concept, the product of individual perception. It is a temporary phenomenon and as such belongs only to the receiver. The difference is not merely one of academic debate. Managements adopting one or the other can affect the design of their Organisations. If information is considered to be a resource then resulting systems are usually more centrally controlled, the assumption being that all information is corporate property. Whereas information considered to be personally owned is seen as being outside the formal structure.

Within the framework of this paradigm the view of information is coloured by its use as a resource. Like any other resource it can be tapped at any time with the certainty of achieving a predictable value from it. Information is regarded as unchanging, and therefore can be easily accommodated into a firm's formal procedure. There are a range of propositions available which seeks to explain information. Each proposition has a consequence for Organisational design. Listed below are those major themes and what their implications might be on business.

Herbert A. Simon took the view that information, along with energy, constitute the two basic currencies of our society. We need energy to breathe, to move, to think, to live. But this alone is not enough. We need to know when to breathe, when to move and how to think. Information provides us with that knowledge. To adopt this perspective is to interpret information as an independent entity. It is all about us waiting to be picked up and used. There may be minor variations in its interpretation by different individuals, but consistency can be easily achieved through better training.

Specific sets of information, such as finance information, are given a unique value by Organisations which are usually based upon departmental rather than individual needs. Particular information is thus attached to particular departments. This allows information to be generally accessible because perceptions of it are bound by the formal departmental framework. Information is then tied into that framework and cannot be used legitimately within any other context, thus guarding its consistency. What is excluded, therefore, is an important role for the individual receiver. This is not to suggest that information cannot be perceived by individuals. Psychologists such as Anderson and Bower have long been aware of the importance of information reception in determining an individual's behaviour.

However, information can be received automatically without an individual having to understand it. For example, through training an employee could process production statistics without knowing what they mean. In this case it is the functional process (through its formal procedure) that is acting as the receiver rather than the individual. An accounting process, for instance, responds to specific inputs without anyone having to understand them. A motor car responds to certain information transmitted through mechanisms such as the throttle, steering wheel or brake. Both of these examples are designed to act as receivers to specific information.

By implication, information can be transmitted to any receiver that logs into the particular transmission. In the same way as a radio picks up a station when it is switched on, the right form will pick up the right information. And like radio waves, information is omnipresent, only needing the correct tuning. Such a view allows information the continuity and consistency required to be regarded as a resource. It also allows a further proposition.

The design of any formal information system must assume that information does not change during transmission. It can, however, be transformed from data or other information sets before transmission. Within such a framework it is the transmitting functions (departments, processes, etc.) which are considered to be the main determinants. The information receiving functions (also departments, processes, etc.) are designed to link into the operational needs of the transmitter functions.

F = functions

Fig. 1. An information network between functions

Each function, however, can be both transmitter and receiver of information, and together with other functions will form part of an integrated network (Figure 1). The functions are tied by their simultaneous roles as information receivers and transmitters, thereby being prevented as much as possible from deviancy in their information usage.

Consequently, business Organisations are driven by those functions which transmit information. The higher the value that a particular set of information is given, the more important will be its relating transmitter function. For example, the finance department could both transmit and receive information. Indeed, each job function within that department could also do the same. A cost clerk, for instance, receives information from the factory floor which he then processes into other information and transmits to some other source, perhaps management. The value of the information he processes will depend upon the utility his Organisation has placed upon it. If the utility is high, then the prestige attached to the transmitting function will also be high irrespective of job complexity. In the same way, if a department has a great number of important information transmitters within its control, then it too will be prestigious and possess influence. Why certain sets of information are prestigious and others are not will depend, amongst other things, upon the Organisation's political processes.

STRUCTURAL PROBLEMS

The difference between these two paradigms may seem subtle, and in terms of information usage it may well be. However, NBS research indicates that the effects upon Organisational design can be major. Structural characteristics consequent upon individually-owned information, for instance, can adopt two extremes. On the one hand they can be organic and informal (adhocracies) whereby all members are seen as contributing equally, each possessing a particular range of techniques to interpret data, whilst at the other extreme there are formal and mechanistic Organisations (bureaucracies). The rigidities of structure stifle any use of information as a resource, so by default such processes are left to the individual.

These contrast with the resource-driven paradigm in which the structural characteristics of Organisations are seen to be centrally orientated, with spheres of influence at the periphery, their importance dependent upon the perceived value of particular sets of information.

ACQUIRED PROBLEMS

The development of an Organisation oriented towards the resource-driven concept of information is well known, and many large firms are aligned in that direction. A sophisticated network is established, more than likely computerized, accessing and distributing information. Particular attention is paid to ensuring standardization of information sets. For example, a piece of costing information is so devised that it can be used by accounts, marketing and production. These Organisations treat all information as their property, attaching to it a value like any other asset. This denies the employee any sense of ownership and tends to detach them from the work process with consequential inefficiencies evolving. It is asserted, that a motivated staff is a contented and efficient staff.

One important way to motivate is to give the individual responsibility within their job. Such responsibility is manifest in how much ability a person will be given to determine the information need of their particular function. To standardize information in the way that many corporate systems do, takes away that fundamental element of responsibility.

Levitan, on the other hand, would claim this to be an inevitable consequence of increasingly sophisticated corporate structures. In addition, Karen Levitan and others imply that information which is not used as a resource is inferior. On the contrary, one could argue that giving the individual ownership could result in a greater entrepreneurial drive.

Standardization is the killer of innovation. It is difficult to conceive of any substantial Organisation, however, where such a degree of individuality could be tolerated. Their size demands a need for cohesion through standardized practice. Therefore it is only in the smaller establishments where such freedom would work. Large Organisations which do not treat their information as a resource still exist.

Their mechanisms of control, however, are not informal or organic. Indeed, the realities are the converse. Individuality is stifled by greater control, rather than less. No-one is trusted to use information for the Organisation's good without formal control procedures. Classic examples of such Organisations are to be found within the civil service or large quasi-government institutions such as the post office.

It has been suggested that there is a relationship between the way in which a firm uses its information and its structure. A matrix can be established differentiating Organisations in terms of how they treat their information and the consequential structural characteristics. If, for example, information is plotted against size (another important contingent of Organisation) four major structural types could be identified. Each is the outcome of the way a particular firm uses its information.

On the vertical axis of the matrix are the two extremes of information. On the horizontal axis are the two extremes of Organisational size (Figure 2). Both information and size are considered to be contingents, that is variables which affect the characteristics of an Organisation. The four possible combinations of these two variables determine four Organisational structures. Thus, each structure is an ideal for each of the four combinations. For example, a managerial structure is the appropriate structure for a large firm using information as a resource.

Resource	Managerial	Technical/ Professional
Treatment of information	Bureaucratic	Entrepreneurial
Perception	***Large***	***Small***

Structure parameters

Managerial	:	larger in size with central/top control but with more departmental integration
Bureaucratic	:	larger in size with formal structure and a diversified control
Technical/professional	:	smaller in size with a professional body and formal structure
Entrepreneurial	:	centrally controlled but with less formal structure and smaller in size

Figure 2: Information/structure matrix

Research into the validity of such a matrix is by no means complete. There is evidence from the NBS regional survey, for example, that such a correlation does indeed exist. In general, the more successful managerial-style firms tend to treat their information as a resource, whilst the more successful entrepreneurial Organisations tend to treat their information as a perceptual phenomenon, and so on. It is not yet certain whether the matrix is predictive, that is implying combinations which are not successful. There is sufficient evidence, however, to support the claim that information usage affects Organisational structure.

The issue of information is much more than whether or not it is a resource. Information is indeed the essence of an Organisation, shaping and determining its structure.

For example, bureaucratic Organisations could hot effectively treat information as a resource because their mechanisms could not control it that way. A technical/ professional Organisation, on the other hand, is better able to handle its information as a resource because of its structural characteristics.

The implication is that there should be a natural balance between the design of an Organisation and its information structure. Where this has not been achieved it is because managers are not matching the structure of their Organisations with an appropriate usage of information. NBS research has also shown that the implementation of computer technology can worsen any potential for imbalance. This is perhaps because of two factors.

Firstly, the characteristics of the technology itself do tend to enhance resource-driven rather than perception-driven information. Secondly, the greater volume of information generated makes the problem more apparent.

There is neither a right nor a wrong way to control information: much depends upon the individual circumstances. Resource-driven information and perception-driven information are opposite extremes of one continuum.

Management teams were grasping eagerly at the new computer technology as a solution to their efficiency problems. There were at that time forces in the environment—market, political, social or economic—with which the traditional methods of Organisation could not cope; it was believed that the computer would provide all the answers. In the space of that time many colleagues began to discover that the impact of computer technology seemed to be contradictory. On the one hand, it did tend to improve efficiency, whilst on the other it was not always so certain that an improvement in effectiveness followed. It seemed that underlying, and as yet not understood, forces were at work.

Departments employing the same technology were more or less successful than others. Business Organisations with all the ingredients of success and implementing computers to further fuel their growth quite often experienced failure a few years later. Since those early days there has been considerable discussion and research on the impact of information technology. It seemed that the perceptual aspects of Organisation were not merely misunderstood but often ignored, and thus we were not able to understand fully the implications of introducing information technology. We now realize that not only does it have an impact on the way we run Organisations, it also has an impact on our behaviour. There is a readily understood physical aspect of any firm and a more abstract, perceptual aspect which arises from the way that people behave as a collective. Information technology is not the only agent in this process, but because of its particular characteristics it can have an extraordinarily powerful effect on both these aspects. One of the two main objectives of this book, therefore, is to underline the importance of the relationship between, information technology (and to a lesser extent general technology) and its Organisational structure.

The two are inseparable. Models that seek to understand either technology or Organisation must do so within the framework created by this relationship. Otherwise an essential factor leading to an appropriate analysis will be missed. The other main objective is to produce a model which can do this. Unlike many of its predecessors, the model will not concentrate on the functional or hierarchical nature of Organisations but rather on the dichotomy between the physical and perceptual aspects. This exists whether or not there is information technology present, but in many cases the implementation of such technology can exacerbate potential dysfunctional forces existing between the two. In effect there are two Organisations—a physical Organisation and a perceptual or virtual Organisation.

The impact that information technology can have on the sometimes tenuous relationship existing between the two can be quite profound. The model explores thoroughly this relationship. It proposes virtual Organisation as the antithesis of physical Organisation. That is abstract, unseeing and existing within the minds of those who form a particular Organisation. The point being that because the framework of virtual Organisation is often subjective and open to many different perceptual interpretations, it is difficult for anyone to establish an appropriate response to it. This makes it no less real; nor does it reduce the need to respond to it. Left unattended the effects of virtual Organisation can be quite catastrophic, especially if stimulated by the impact of information technology. This has considerable significance for the practitioner, especially those who have introduced, or are thinking of introducing, information technology. The model suggests that for a firm to survive there would ideally exist a dynamic equilibrium between the two Organisations which has been termed the Organisational balance. Anything which upsets this balance, such as the introduction of information technology, has to be compensated for by proactive action of management. If managers do not understand, nor indeed perceive the existence of virtual Organisation, then they will not be able to take the appropriate measures. Long-term structural weaknesses or even total failure of the business can result.

Demonstrating the existence of virtual Organisation can be problematic. By definition it cannot be observed directly since it exists only in the mind. At best one could ask a selection of staff for their impressions and make certain deductions, but these are likely to be dubious and tainted with the researcher's own perceptions. As such there has deliberately been no direct empirical research to establish the foundations of this model. Support for its validity has been sought in three other areas. Firstly, there is indirect empirical data gained from several research projects conducted at the Newcastle Business School. This comprises a study over two years by Stuart Maguire and myself into eighty-five small to large firms in all sectors of the North East. We examined particularly the strategy that firms employed to implement and control their information technology.

The data were gathered through a series of observations, interviews and involvement in the process. Other data were obtained from one or two firm studies on the impact of information technology by a group within the Newcastle Business School called AMBIS (Advanced Manufacturing and Business Information Systems). The argument for using these data rather than dedicated data is that they should be less contrived. Secondly, there is support to be gained from relevant personal experience. Although many would find this approach invalid because it does not comply with traditional or conventional ideas about research data, there is an increasing acceptance that such data can prove useful. In the social sciences, for example, Duke argued that such an approach (amongst others) would make a valuable contribution to the understanding of social experience. It is accepted that it cannot provide the only data, but the data that it does provide should be considered no less valid than any other.

We return to a more traditional style in support of the main arguments. That is, rather than gathering empirical evidence first and then developing a model, we should develop the model first from experiential instincts and then support it with data. This is put forward in the belief that to pursue the traditional methods of data-first research would not produce the generic model necessary to achieve the objectives set above. We have become unimaginative in our ideas about Organisation because we have been blinkered by too many sets of data.

How the book is structured in terms of presenting a logical argument takes on a greater significance than if it were based merely upon direct empirical data. It requires presentation of basic argument, counter-argument and a compromise position which leads the reader forward to the logical conclusion. This is the dialectic which involves thesis, antithesis and synthesis.

This overview, therefore, should not only cover the background to the book's major arguments and supportive material, but also provide the reader with a basic chapter plan.

To explain the impact of information technology we must look beyond the functional aspects of business Organisations.

The importance of information is reflected in its ability, through the appropriate technology, to corrupt departmental boundaries. This chapter, therefore, proposes that it is better to understand an Organisation as a network of information flows, which is itself set within other network flows of region, market, industry, and so on. Such a perspective is underlined by many firms accessing information beyond their established functional boundaries. Within the Organisation, this could involve different departments accessing a common pool of information from the company database.

Externally, employees can gain entrance to public databases such as Prestel, which are run by autonomous bodies. The drive is towards centralization and hence better management control. Paradoxically, the opposite is being encouraged, whereby every user has the potential to access information which enables them to operate more efficiently, and also to do so outside the control of their manager. Under these conditions it is difficult for managers to control not only who reads and writes what but also, and perhaps more importantly, who knows what.

Large amounts of information can be processed and communicated via a network without managers ever being able to determine whether it runs contrary to their own policy. Security has, as a consequence, become an extremely important issue. The more the dissemination of information is allowed, the more likely control mechanisms will become corrupted.

In simple terms, data are facts, whilst information is the consequence of interpretation. What is information to some is not to others, and it is the subjective interpretation of what is and is not valuable which has caused managements considerable problems. The control parameters on the Organisation's database are based upon managers' own perceptions of its value as information which may not coincide with that of a potential intruder or misuser. But more than this, individual employees' perception of their information stock widens because of their greater accessibility to the information network. As a consequence, their perception of their own job parameters also widens beyond what management might think desirable.

NEW EXAMPLES

The perception-driven paradigm does not consider information to be a resource. Individuals or groups are seen to own their information. What belongs to the Organisation are the data sets, the facts and figures specific to certain functions. For instance, data on how an Organisation is performing in its market quite obviously belong to that firm. How well that data is used is dependent upon the individual's own competence in interpreting it.

Management can control this indirectly through training, but they cannot directly control the thought processes which turns data into information. Once again, there is a collection of propositions or ideas about information which formulate a general approach.

Adherence to this paradigm, as in the previous paradigm, may not necessarily be by conviction but rather by default because people not treating information as a resource automatically form part of this group.

However, there are others who are members by conviction. Information is said to be receiver-dependent, in that, any set of information can be considered to be information only if it is recognised as having value by a particular individual. Thus, a river flooding may be information to someone who lives on its banks but of no consequence to someone living several miles away.

It would be difficult, therefore, to understand information as a group or Organisations property since its determination is subjective. Management teams have implemented computer technology and in so doing also developed their information sets into a resource. They experience problems, usually some time after a successful launch, and then abandon using their technology in an integrative manner, returning to the 'islands' approach (that is, a computer system, perhaps physically integrated through networks, etc., but perceptually segregated into different functional areas). Information is no longer seen as a resource; indeed its treatment as such is often seen as the culprit and not the technology itself.

Both paradigms recognize a difference between data and information. Computer personnel, for example, have long known of the distinction. Information is seen by them as processed data. The two are interrelated, data being the input to a process, and information the output. In the same way that an output of one system can form the input of another, information from one system can form the data of another.

This is consistent with treating information as a resource. Perceptually-conceived information on the other hand is not the consequence of a formalized process. Data are a factor, but then so are other aspects such as individual traits, culture, structure and political processes.

The relationship between data and information is, therefore, not so strong. This contradicts the objectives implied by the creation of management information systems and data processing. The former allows the individual access to established information, whilst the latter converts data into information. Since the production of information cannot be formalized there is no point in trying to achieve it. Systems should, therefore, be designed to produce the most accurate and readable data in order to enhance easy conversion into information.

When Ligomenides describes information as a fundamental force, he should be referring to data. Levitan, on the other hand, is correct in claiming information to be dynamic and continuously evolving. The logic of these two views can be seen when data and information are differentiated. It is only data which is time-independent, and therefore unchanging and static— a fundamental force. By this we mean that data retains a constant empirical value: a fact is a fact and cannot be altered. Information, on the other hand, can be altered and changed. It is time-dependent and thus its value can alter from one moment to the next. It is dynamic and evolving because it is determined by an individual's perception.

It is, indeed, time and utility which transform data into information. If at a particular time a set of data is useful to an individual then it can be described as information. Equally, at another time the same set of data may not have utility and therefore not be information. The data themselves have not changed in any way, remaining always constant. What has changed is the individual's perception of those data. Information is in reality the consequence of a complex psychological process which transforms perceived data, into usable thought inputs.

It is, therefore, data which the individual receives from the environment, the brain which transforms these data into information. No formal system can directly pre-empt this stage; thus no formal system can transform data into information.

Data can be transformed into different data, transmitted and received. For example, production data submitted to the cost department and processed into second generation costed production data, which are in turn submitted to the financial director. Information is transformed, in part, from data; it can only be received and never transmitted. The financial director may attach value to costed production data and thus receive it as information. If he/she passes it on, it will once again be as data, perhaps refined into third generation data. There cannot be information systems because information is so transient.

The individual's subjectivity, which can be developed by training and formal procedure, determines what is information.

At another time that perception may be different. Information systems cannot work in this way. Their assumption must be that information is not transient but consistent.

Quite obviously information systems do exist in one form or another. They may not be what they claim to be but they are nevertheless working. Is it worth arguing the difference? The answer to that lies within ourselves. Our actions and behaviour, and consequently our Organisations, are governed by our perceptions. If, therefore, we understand information to be permanent rather than transient then how we organize ourselves will be coloured by that. We will design systems which use information as a resource and expect it to be totally consistent.

Four

LEARNING TOOLS

A major segment of the market involves entertainment: personal computers offer a growing array of games, music and art far more sophisticated, in terms of speed and capability, than video material based only on television. This advantage results from the ability of computers to store and quickly retrieve large amounts of information. The future will bring many more uses for computers in both business and home. Indeed, the possibilities are virtually endless. A great variety of computers, large and small, are now available. A personal computer is a machine or system meeting all the major qualifications.

The system is designed to accept secondary memory devices to supplement the primary, built-in memory, the user is expected to interact with the system continuously, not only at the beginning and end of a problem, at least one general language (Basic, Fortran, Cobol, Pascal, ADA, or Q is available for this interaction), the system is usable for a wide variety of problems and is not designed for any single application and the computer is distributed through mass-marketing channels, with the marketing emphasis on the first-time computer user. A typical modern personal computer consists of a circuit board with a silicon chip, microprocessor and one or more memory chips attached. The microprocessor can perform hundreds of thousands of calculations every second, and the memory chips provide the primary storage for instructions and data.

External storage devices, such as cassette tape units or small recording disks (floppy disks), augment the memory capacity and provide a storage medium that can be physically transferred from one personal computer to another. (Typical users begin with cassette units but soon change to disks to gain the advantages of greater speed and capacity.) Input is through a typewriter-like keyboard unit.

Output typically takes the form of words and numbers displayed either on a television screen or a similar specialized screen called a monitor. Most monitors are designed to display 24 lines of letters and figures, with each line containing a maximum of 80 characters. Adding a printer unit permits output in the form of a printed paper. A special device, called a "modem" (for modulator/demodulator), permits the computer to receive and transmit data over a conventional telephone line. But the significant factor in defining a personal computer is not its physical features but the characteristics of the operating system.

Designers of personal computers and software attempt to provide a friendly human-machine interface, even at the expense of brute computing power. Optional programmes are also available so that the computer can be used for many different purposes. Although word processors and hobby computers have many characteristics of personal computers, they lack this flexibility.

Recent trends in microelectronics, memories, input-output mechanisms, and software suggest that the trend in microprocessors is towards larger "words" and higher circuit speeds. A computer capable of handling larger "words" is able to perform a complete operation in fewer machine cycles and to operate directly with larger memory. Both these assets enhance performance by increasing the speed of operations and the number that can be performed in a sequence without the operator's intervention. In addition to greater speed and memory access, these larger microprocessors have the advantage of greater accuracy.

The first wave of personal computers used 8-bit microprocessors—that is, microprocessors in which 8 binary digits (0 or 1) can be processed in parallel, giving the capacity to process in a single operation any number up to 256, or to address any file up to 256 points in the memory. Processing larger numbers with an 8-bit microprocessor requires multiple operations, which take more time. Newer systems use 16-bit microprocessors, and 32-bit microprocessors are now available.

The primary memory in personal computers is of two different types: read-only memory (ROM) and random-access memory (RAM). In the former information is fixed in the memory at the time of manufacture and is not lost when computer power is switched off. The role of such a ROM is to guide the computer through a fixed procedure, such as calculating square roots and translating a user's programme into machine language.

In a RAM, information such as special programmes and data files can be "written in" or "read out" as frequently as desired, with any storage location directly accessible. RAMs are of two types: dynamic RAMs, which are cheaper but lose their store information unless they are "refreshed" often, and static RAMs, which are costlier but do not need to be "refreshed." If power is lost, both types of RAMs lose their stored information.

Over the past decade, the number of memory circuits per unit of area on a chip has increased by a factor of 64, and cost on a unit basis has been reduced by a factor of 50. Both these trends will continue. So-called 64K dynamic RAMs (each contains 65,536 bits of information [K = 210 = 1024; 64K = 64 x 10241) remained popular until about 1984, when the much larger memory capacity of 256K dynamic RAMs became standard.

Most manufacturers supply system programmes on ROMs, and this practice will continue because programmes are secure against power failures and users are less able to duplicate programmes in this form. Computers' main memories will be supplemented by secondary storage devices that offer larger and relatively inexpensive, though slower, capacity for long-term storage of programme and data files. The most popular of these is a so-called "floppy disk"-a disk of mylar coated with magnetic material on one or both sides. Data are stored in a series of spots-either magnetized or demagnetized-along the concentric tracks. Heads for reading or writing data can be moved radially across the disk to reach a specified segment of circular track.

Storage capacity depends on the format used for the stored data, the quality of magnetic surface, and the design of the reading-writing head. Floppy disks in current use typically have capacities of I to 4 million bits, sufficient to store 20,000 to 80,000 words of English text. During the next four years, higher-density floppy disks will be common. Indeed, disks offering capacities of 50 to 100 million bits are already becoming popular, but they are much costlier than floppy disks.

The primary display device used in all personal computer systems is a cathode-ray tube (CRT), either standing alone or as part of a television receiver. This system will continue through the foreseeable future. Output is typically presented in alphanumeric form-letters and numbers. Charts and game boards can be presented, but the memory and software required to display such graphical images is often complex and expensive. The letters and numbers in alphanumeric displays are patterns of dots programmed in special ROMs known as "character generators." The quality of the image depends on the number of points (or pixels) on the screen that can be addressed by the computer—that is, the number of points at which dots can be located. A typical low-resolution screen has a field of 6144 (128 times 48) pixels (picture elements-dots on the screen). High-resolution systems (100,000 pixels or more) allow sophisticated graphics for animation or detailed figures and may provide colour as well.

PERSONAL COMPUTER

No one should select a personal computer system without taking into account the tasks for which it will be used and the environment in which it will function. Even within a single business, personal computer users come from different areas—personnel, accounting, management, manufacturing, research, sales—and have different computing needs.

Even within these departments there are many possible uses. The personnel department will run personnel files on a regular basis but also may want to study policy questions—the likely benefits in terms of increased revenue from a larger sales force, for example. Furthermore, different users have different strategies for using a computer for ad hoc applications. Thus, the person who will use a personal computer should—almost by definition—choose it, partly on the basis of the services it will provide and partly on the basis of its apparent user-friendliness.

Indeed, the machine's responsiveness to its user's needs and style is the critical technological breakthrough. But it is not necessarily so simple. In many Organisations, personal computers create and fill completely new needs that were not originally anticipated. Consider the example of a personal computer acquired to improve one lay worker's access to a main computer facility. While retrieving data from the main computer to run programmes locally, this user finds that the personal computer can help evaluate alternative business strategies, answering, "what-if...?" questions instantaneously. The original application for which the personal computer was purchased becomes secondary. Given that the uses and benefits of personal computers are hard to predict, and that a wide range of systems and software are available, how should a potential buyer decide what and when to buy? And should such a buyer postpone action, expecting costs to decline in the future as they have in the past?

There are two arguments against postponing purchase in anticipation of a lower future price. One is the high cost of waiting, based on the computer's great (and often unexpected) uses. The other is that prices are expected to decline less dramatically in the future than in the past: future systems will offer technological improvements that increase performance at unchanged cost. Prices seem likely to stabilize at around $1500 for a full personal computer system.

Future personal computers are likely to offer increased memory size, increased processor power (16-bit and even 32-bit instead of 8-bit microprocessors), improved printers, more powerful programming languages that give the user more compact and natural communication with the computer, improved user access (better keyboards supplemented or replaced by touch screen and voice systems), and greater flexibility (the computer will be able to intermix and manipulate text, numbers, and graphics more easily). Yet today's personal computers may rapidly become obsolete, reflecting the rapid pace of change in semi-conductor technology.

Changes in the microprocessor, the heart of any personal computer system, are most likely—and most serious. A computer with a 16-bit microprocessor cannot operate on hardware and software designed for an 8-bit microprocessor. Though many vendors provide add-ons that enable a new microprocessor to support old software, there is never total compatibility.

Software for a personal computer is expensive, and for most users new software development is impractical. So the amount and utility of the software available with a particular system is the key question in the buyer's choice of a personal computer. The variation can be large: 11,000 programmes are available for the Apple II and Apple II Plus computers, whereas fewer than 100 widely distributed programmes exist for the IBM personal computer, a more recent introduction. Of course, the IBM programmes may be just the ones the buyer needs, and more software will be developed for the IBM.

Price is also an important variable, including not only the manufacturer's suggested price but the range of discounts offered by different retail computer sources. But discount alone does not tell the story. The critical factor is the ability of the distributor to provide both routine hardware maintenance, usually at the retail outlet, and software support and maintenance. A company store is likely to offer very good service on that company's product line but not necessarily on any other line, while a retail computer store is likely to offer moderately qualified service and support for many different manufacturers' systems. Mail order sales offer very little service but usually have the highest discounts.

The ultimate question for the purchaser is that of system configuration: what combination of hardware and software is appropriate for a buyer's current and projected needs? Perhaps the most important options are off-the-shelf software. To be sure they buy what they need, buyers should gain some experience with available programmes before making the purchase; otherwise they risk having to modify their needs to fit the software they've purchased or embarking on the costly and time-consuming route of software development.

The programming tools on most personal computers are primitive compared with the equivalents available for mainframes, and the widespread dearth of good programmers is a complicating factor. Indeed, trying to develop special programming for a personal computer is inadvisable except under extraordinary circumstances. Memory capacity is another important variable. Most computers based on 8-bit microprocessors have no more than 48 or 64 kilobytes of built-in accessible memory (RAM), while the newer 16-bit processors can have 1, 4, or even 16 megabytes. (One byte is a string of 8 binary digits that represents one alphabetic character.)

Additional memory can be inserted into most personal computers by plugging cards into slots or inserting extra chips directly onto the microprocessor board. Storage capacity of the main memory for programmes and instructions can also be supplemented by secondary devices, of which floppy disks are the most popular form. Disks with capacities ranging from 100,000 bytes to several megabytes now come in two standard sizes: 8 inches and 51/4 inches in diameter. But personal computer buyers should be aware that disks designed for one system will not generally operate in another maker's system because of different data-formatting conventions.

The critical performance characteristic of a floppy disk is the response time to a read-write request, which may range from 200 to 500 milliseconds. Though the response time for such a single file request is very short, a single compilation or word-processing command may involve 20 or more accesses to a floppy disk, and access time may account for 80 percent of the total command execution time. Where better performance-and/or capacity are required, "hard" disks in the same format as the 51/4-inch and 8-inch floppy disks are useful. They offer storage capacities ranging from 5 to 50 megabytes and response time averaging 20 to 40 milliseconds, but their high cost (around $2000, including interfaces) can double a buyer's total investment in hardware.

All personal computer systems use cathode-ray tubes as output devices. If "hard copy" is required, buyers have a choice of several printing devices. Thermal printers, costing under $500, create images by applying points of heat to special paper. A dot-matrix printer, costing from $500 to $1500, has a vertical array of 7 to 10 dot-printing elements. These are activated by the computer as the printing head passes across the paper, forming alphanumeric characters and also graphs and drawings.

The limited number of elements result in a somewhat stylized character, but the process is as fast as thermal printing—50 characters per second. If letter-quality printing is required, a printing mechanism with precisely machined character matrices that strike the paper through a ribbon is required. Such printers cost $750 or more and print between 30 and 90 characters per second. Various accessories such as floppy disks and printers or "peripherals" are attached to personal computers through special "slot" receptacles. The IBM personal computer offers five such slots for peripherals such as printer, colour display, dual floppy-disk controller, communications channel, and memory module.

All business-oriented personal computers offer text editing for word processing. Some permit use of a light pen to directly indicate on the screen the word to be edited, and a few check spelling against a mini-dictionary stored in the computer memory. Business data manipulation is facilitated by "spreadsheet" packages such as VISICALC—a very popular package that enables users to see on the CRT the impact of altering one figure (say, retail price) on all other figures (profit margin, corporate profit, return of investment).

Software is now becoming available for generating displays such as those accompanying this article with only a few minutes' effort. For example, Execuvision permits displays combining infirmation in numbers, text, and pictures. It also provides animation that can add emphasis to presentations. Though personal computers are today viewed primarily as stand-alone work stations for individual users, the ability to transmit and receive data and programmes is an important attribute of most contemporary systems. Personal computers are already serving as intelligent terminals for retrieving and manipulating files from large computers. These networking and communication capabilities are important considerations in selecting a personal computer. Unfortunately, no industry standard for communication protocols and interfaces exists. Different vendors use different protocols, so it is difficult to link various personal computers in a single network.

Technological advances in communications are expanding the power of personal computers. For example, owners of personal computers can now receive current stock market quotations by telephone. Personal computers can be connected with nationwide electronic funds transfer systems, so users have access to full banking facilities. They may transfer funds from one account to another and buy and sell financial instruments.

In homes, personal computers are now primarily used for recreational activities such as computer games. But these applications should not be underrated. Computer games test and develop mental capabilities while generating familiarity with computer operations. Home computers are also used for office work at home, and many other applications are, or soon will be possible. For example, a tax return can be prepared with the aid of VISICALC. The user can analyse which forms and deductions to use for paying the minimum taxes. A personal computer can be used to balance check-books and plan investment strategies. And, of course, word-processing capabilities are useful in writing letters and professional papers.

As commercial databases such as Dow Jones News/Retrieval and the Source are expanded and their subscription costs reduced, personal computers will be useful for locating stores and services, comparing prices, and placing orders. This would be especially appropriate for products or services whose prices fluctuate rapidly, such as those of the airline industry. Computers are being used increasingly for security applications. Traditional home security systems are limited, often failing to distinguish between a natural event and an intrusion by an unwanted visitor. Computer- controlled home security systems can include a variety of routines to analyse the signals from sensors before generating an alarm. Such an analysis can distinguish, for example, between the entry of a cat and the entry of a human being, or between the noise of a telephone ringing and a door opening.

Another home-oriented application is a medical information system that performs some of the diagnostics of a medical doctor. Perhaps the most useful application of this is a database for poison or first-aid: the user reports what substance the patient has ingested or the nature of an accident, and the computer responds with a step-by-step first aid plan. Children turn out to be heavy users of home computers. They enjoy computer games and seem able to learn computer languages faster than their elders. They find personal computers intriguing for recreation, school assignments, and self-paced courses that are now becoming available. Even though today's personal computers are equivalent in basic computing power to the mainframes of the 1960s and the minicomputers of the 1970s, they should not be used as substitutes for the earlier machines. Rather, users should capitalize on the new technical developments, including the diminishing cost of computing, and the English-like languages that replace the cumbersome digital languages of early machines, to make personal computers supplement, not displace, mainframe computers.

Personal computers should be used to analyse more issues in greater detail than ever before—not to do the more routine work typically assigned to mainframes. In the 1960s and 1970s, computers were used principally for well-structured, periodic jobs such as payroll accounting. Today's personal computers are especially useful for more casual, ad hoc analyses and problem solving, such as studies of the impact of cost of living increases, new tax regulations, or alternative hiring and layoff policies.

Consider the stock analyst who uses data from company balance sheets to evaluate investment risks. The balance sheets contain detailed, current information on assets and liabilities. But interest rates fluctuate, markets rise and fall, taxes change, competition falters or gains. Each such change affects a company's current performance and future prospects.

Using a central computing facility is an expensive way to analyse such problems, resulting in time delays and partial answers. But using a personal computer in the interactive mode linked to the central computer and a financial database makes it possible to understand the impact of changes quickly and accurately.

WORKING BEHAVIOUR

Manufacturers, media, and computer scientists have presented the public to computers the way one might have presented Aladdin to his lamp. Here was the genie, the workhorse that we could mount today and ride onto the new millennium. By the end of the 1970s, with the mass manufacture of personal computers, the promises extended into the home. Home computers would teach us French, help us with financial planning, even do our taxes. For many Americans, the first thing that computers brought into their homes was not a more efficient medium for work but new worlds to conquer in play.

"Space Invaders" became a household word for people with home computers or the machines' junior siblings, computerized video games. The experience of the decade has resulted in a widely shared public rhetoric. When people talk about the computer and their futures, they tend to fall back on the two images they know best: the computer as tool and the computer as toy; objective instrumentality and engrossing play. But there is another dimension. What people do with computers weaves itself into the way they see the world. People use a discourse about computers for thinking and talking about other things, about politics, religion, education, and about themselves and other people.

The subjective dimension of the computer presence is not merely a matter of discourse, of using the computer intellectually as a metaphor or model. There is another, and more emotionally charged, aspect which does not engage ideas about computers as much as the immediate quality of an individual's experience when working with them. Working with computers can be a way of "working through" powerful feelings. In this essay we develop this idea by using as a case study a group of computer users for whom issues related to control are particularly salient. For a first generation of computer hobbyists, controlling the computer is a way to deal with frustrations and desires, both personal and political, that have nothing to do with the computer per se.

The subjects of my study are men and women who bought personal computer systems in the four years that followed the 1975 announcement of the "Altair"—the first computer small enough to sit on a desktop, powerful enough to support high-level language programming, and that you could build for only $420. My study began in 1978 with a questionnaire survey answered by 95 New England computer hobbyists (their names had been drawn from the roster of a home computer club and from the subscription list of a personal computer magazine), and continued during 1978 and 1979 with nearly 300 hours of conversation with 50 individuals who owned home computers. What we found can be read historically: a study of the pioneer users of an increasingly ubiquitous technology. But most central to the intent of this essay is to use the story of the early hobbyists as a window into the highly personal ways in which individuals appropriate technologies. It is a case study of the "subjective computer," the computer as a material for thinking, for feeling, for "working through."

It is acknowledged that the subjective is at odds with a widespread ideology that quickly grew up around the emergent computer hobbyist culture. The Altair, aimed at a strictly hobby market, was followed by other small systems—the Pet, the Sol, and, most successfully, the Radio Shack TRS-80 and the Apple, marketed to less specialized audiences of small businessmen and curious householders. With this explosion of hardware came a lot of rhetoric about a personal computer revolution. Most of the talk, both from the companies that marketed the machines and from those who claimed to be the most visionary spokesmen for the people who bought them, was about all of the things that a home computer could do for you.

The utilitarian, "genie in the bottle," ideology is expressed in the content of hobbyist conventions and magazines, filled with articles on how to make your home computer dim your lights, control your thermostat, run an inventory system for your kitchen or tool-room. And it is also found in writing on the personal computer from outside the hobbyist world. The view from the "outside" was well illustrated when, on May 14, 1979, the *Wall Street Journal* reported on its own little "evaluation experiment." The paper had drafted one of its staff reporters, Mitchell Lynch, into the ranks of the "home computer revolution." It presented him with a TRS-80 and asked him to take it home. His assignment was to report back six months later on what it had been like.

The terms of Lynch's evaluation are instrumental: what can the computer do? On these terms it fails, and when he goes to the experts for an explanation of what went wrong, he encounters an official "instrument" ideology from within: "Experts say that people like me have neither the technical training nor technical inclination to make a home computer strut its stuff." The point of having the computer, says the expert, is to make things happen. And, reassures the expert, they will happen with a better operator or a simpler computer. This instrumental view is an important ingredient of the hobbyist ideology but it is not the whole story. In the course of my work I found a very different answer from within.

Most hobbyists do make their computers "strut their stuff," but their sense of engagement and energy are found primarily in the non-instrumental uses of the technology. When asked in a questionnaire "What first attracted you to computers?" more than half the respondents gave reasons that were highly subjective. In response to an open-ended question, 26 per cent said that they were first attracted to computers by an appeal that was intellectual, aesthetic, involved with the fun of what I would call "cognitive play." They wrote of "puzzle solving," of "the elegance of using computer techniques to handle problems," of the "beauty of understanding a system at many levels of complexity." They described what they did with their home computers with metaphors like "mind stretching" and "using the computer's software to understand my wetware."

Another 26 per cent wrote of reasons for getting involved that seemed more emotional than intellectual. They wrote of the "ego boost" or "sense of power" that comes from knowing how to run a computer, of the "prestige of being a pioneer in a developing field," of the "feeling of control when I work in a safe environment of my own creation." The hobbyists who responded to my survey seemed familiar with Lynch's brand of scepticism, with people who ask them what they do with their computers and who won't take "cognitive play" for an answer. David, a 19 year-old undergraduate at a small engineering school, put it this way: "People come over and see my computer and they look at it, then they look at me, then they ask me what useful thing we do with it, like does it wash floors, clean laundry or do my income tax-when we respond no, they lose interest." David said that when he started out, he was attracted to the computer because "we liked the idea of making a pile of hardware do something useful, like doing real time data processing ... like picking up morse code with an amateur radio and transcribing it automatically into text," but in his list of things that he currently does with his computer.

Thirteen percent of those who responded to my questionnaire told a similar story. Like David, they began their relationship with personal computation for instrumental reasons (they had an image of a job to do, a specific task), but they became absorbed by the "holding power" of something else. A full two-thirds of my survey sample either began with or ended up with a primary interest in what I have called the "subjective computer," the computer seen in its relationship to personal meaning. Clearly, to understand what people are doing with their home computers we must go beyond the "performance criteria" shared by the hobbyist magazines and the *Wall Street Journal*. The simplest way of thinking about the subjective computer is through the metaphor of "the computer as Rorschach"-that is, seeing the computer as a projective screen for other concerns. In the Rorschach test, one is presented with a set of ink blots and asked to make some sense of them. Some people see the blots as threatening; others see them as benign. Some focus on small, complex details; others on global form. So too, the computer presents us with an ambiguous stimulus. For example, although most people think of the computer as an object without real intentionality, they accord computers enough autonomy in action to make "blaming the computer" a commonplace of daily life.

The very fact that the computer is a machine that touches on a sphere like intelligence that man has always considered uniquely his, is enough to make many people experience the computer as an object "betwixt and between," hard to classify, hard to pin down. Sometimes people deny the irreducibility of computation by asserting that, no matter how complex the computation "product," a move in a chess game for example, "all the computer really does is add." Of course in a certain sense this is correct. But saying that a computer "decided to move the Queen by adding" is a little bit like saying that Picasso "created Guernica by making brush-strokes."

Reducing things to this level of localness gives no satisfying way to grasp the whole. Just as in theoretical psychology there is a tension between the gestalt and the atomic, so too in computation there is a pervasive tension between the local simplicity of the individual acts that comprise a programme or a computation, and what one might call the "global complexity" that can emerge when it is run. The elusiveness of computational processes and of simple descriptions of the computer's essential nature, the tension between local simplicity and global complexity, all contribute to making the computer an object of projective processes, and exemplary "constructed object." Different people apprehend it with very different descriptions and invest it with very different attributes. In views of the computer's internal process, individuals project their models of mind.

In descriptions of the computer's powers, people express feelings about their own intellectual, social, and political power, or their lack of it. Looking at the computer as Rorschach, has projected, puts the emphasis on aspects of the individual—from cognitive style to personal fears—that are revealed through behaviour with the machine. But of course the computer is more than a Rorschach. The Rorschach ink blots are evocative, revealing, but they stay on the page. They do not enter the life of the individual. The computer does. It is a constructive as well as a projective medium. For readers who have not had the experience of programming a computer, this idea may be sharpened by an analogy with another technology. In my own studies of people's emotional relationships with technologies, aeroplanes emerge as startlingly like computers in respect to the issues they raise for their hobbyists. Specifically, both are powerful media for working through the issue of control.

MANAGING TOOLS

There is always a compelling tension between local simplicity and global complexity in the working of a computer and in the appreciation of a computer programme. Locally, each step in a programme is easy to understand; its effects are well defined. But the evolution of the global pattern is often not graspable. You are dealing with a system that surprises. This play between simplicity and complexity allows programmers, as pilots do with flying, to make of computation very different experiences that provide a context for working through different needs in relation to issues of personal control. Depending on how the programmer brings the computer's local simplicity and global complexity into focus, he or she will have a particular experience of the machine as completely understandable, under control, or as baffling, even as controlling. By focusing on the local, the line by line, you can feel in control. By focusing on the global, you can feel control slip away. In their style of programming a computer, people betray different levels of tolerance for temporary losses of control.

Some will avoid it at all costs; but others will seek it out, and enjoy "playing" with sensations of risk and danger. And so different people end up with very different relationships to controls and power in their programming work. To illustrate this point, it is useful to begin with an example at one extreme. We see a first style in Howard, an ex-programmer, now a university professor, who describes himself as "having been a computer hacker." Howard was not in my sample of hobbyists. He has a terminal at home which links to a large timesharing system, but when asked about home computers, he winced in distaste and said that he "wouldn't touch the stuff. It's too simple."

His case, in which we see a love of programming for the feeling of "walking near the edge of a cliff," is meant as a contrast for what we found to be a prevalent and more conservative "hobbyist style." Howard described his longtime fantasy that he would walk up to any programme, however complex, and "fix it, bend it to my will." As he described his intervention, he imitated the kind of hand gestures that a stage musician makes towards the hat before he pulls out the rabbit.

Wizards use spells, a powerful kind of local magic. Howard's magic was local too. He described his "hacker's approach" to any problem as a search for the "quick' and dirty fix." For Howard, what was most thrilling about the experience of programming was "walking down a narrow line," using the program's flexibility (for him defined as the possibility of making a local fix) in a struggle to keep the whole under control. Weekends at the terminal with little to eat and little or no rest were frequent, as was the experience of not being able to leave the terminal while debugging a programme, even when the obvious need was for sleep and looking at the whole in the morning, instead of trying to "fix it" by looking at it line by line all night. For Howard, the urgency of these encounters was tied to his sense that through them he was grappling with a computational essence—the struggle to exert control over global complexity by mastery of local simplicity.

A second programmer, Bob, is a computer professional, a microprocessor engineer who works all day on the development of hardware for a large industrial data system. He has recently built a small computer system for his home and devotes much of his leisure time to programming it. Whereas, for Howard, the excitement of programming is that of a high-risk venture, Bob likes it as a chance to be in complete control. Although Bob works all day with computers, his building and programming them at home is not more of the same.

At work he sees himself as part of a process that he cannot see and over which he feels no mastery or ownership: "Like they say, we are just a cog." At home Bob works on well defined projects of his own choosing, projects whose beginning, middle, and end are all under his control. He describes the home projects as a compensation for the alienation of his job. He works most intensively on his home system when he feels furthest away from any understanding of "how the whole thing fits together at work."

Howard and Bob have very different opinions about what is most satisfying about programming. These translate into different choices of projects, into different choices of programming language and level to programme at, and ultimately into what we might call different computational aesthetics. Howard likes to work on large, "almost out of control" projects. Bob likes to work on very precisely defined ones. Howard finds documentation a burdensome and unwelcome constraint; Bob enjoys documentation, he likes to have a clear, unambiguous record of what he has mastered. Indeed, much of his sense of power over the programme derives from its precise specifications and from his continual attempts to enlarge the sphere of the program's local simplicity.

Hobbyist programmers mean different things when they say that machine language programming "puts them in control." For some, the reference seems objective. Given the primitive higher-level languages available on the first and second generation of hobby computers, machine-level programming seems to them the best instrumental solution. For others, the issue is more subjective. Many hobbyists who said, they felt Lineasy using systems programmes for which they didn't have the source code also admitted never looking at the source code listing which they felt they had to have. Having access to the code was symbolic.

In other cases it was apparent that machine language programming was valued because the experience of doing it was pleasing in itself. It meant that the programmer was writing instructions that acted directly on the machine, no "building on top of" somebody else's interpreter. Most hobbyists have relationships with computation at work which involve sharing the machine with countless and nameless others. In personal computation they see a chance to be independent and alone.

The machine comes to them virgin. They have full possession. Finally, there is the issue of asserting control over an inferior. Bob, like many of the other hobbyists I spoke with, is a middle-level worker in the computer industry. He does not feel very good about the importance of his job. Proving that he is "better than any dumb compiler" gives a sense of importance. Using the computer to assert control was a central theme in my interviews with hobbyists. It was expressed directly, and also wove itself into four other issues that characterize the hobbyists' "subjective computer." These are using the computer to strengthen a sense of identity; to construct a completely intelligible piece of reality that is experienced as "transparent" and safe; to articulate a political ideology; and to experience a sense of wholeness that is absent in one's work life. It is these four issues to which I now turn.

BRAND OF MANUFACTURE

In achieving a sense of mastery over the computer, in learning about the computer's "innards," people are learning to see themselves differently. Among other things, they are learning to see themselves as "the kind of people who can do science and maths." This was most striking among hobbyists who had no technical background. But it also came up among hobbyists who did see themselves as "technical people" but who for one reason or another had gotten "scared out of real science." Barry is 28 years old, an electronics technician at a large research laboratory. He went to college for two years, hoping to be an engineer, then dropped out and went to technical school. He has always loved to tinker with machines and to build things. His current job is to calibrate and repair complex instruments, and he is very happy with it because he gets a chance "to work on a lot of different equipment." But he came to his job with a feeling of having failed, of not being "analytic," "theoretical," of not being capable of "what is really important in science."

Five years ago, Barry bought a programmable calculator and started "fooling around with it and with numbers the way I have never been able to fool around before," and says that "it seemed natural to start working with computers as soon as I could." To hear him tell it, numbers stopped being theoretical, they became concrete, practical and playful, something he could tinker with. I'll pick up the calculator, and if we don't know how to do a problem I'll play with the calculator a few minutes, or a few hours and figure it out. It's not so much that the calculator does a particular calculation, but you do so many, have so much contact with the numbers and the results and how it all comes out that you start to see things differently ... The numbers are in your fingers.

When the calculator and the computer made numbers seem concrete, the numbers became "like him," and Barry felt an access to a kind of thinking that he had always felt "constitutionally" shut out of: "When I write in assembler I feel that mathematics is in my hands ... and I'm good with my hands." Barry claims to have "grown out of" his aspiration to be an engineer. He says he doesn't keep engineering as a pipe-dream or think of his computer skills as something that could make it real. In terms of his career he says that "nothing has changed." But a lot has changed. Barry has always thought of himself as a bundle of aptitudes and inaptitudes that define him as the kind of person who can do certain things and cannot do others.

Working with the computer has made him reconsider his categories. We really couldn't tell you what sort of thing I'm going to be doing with my computer in six months. It used to be that I could tell you exactly what we would be thinking about in six months. But the thing with this, with the computer, is that the deeper you get into it, there's no way an individual can say what he'll be thinking in six months, what we are going to be doing. But we honestly feel that it's going to be great. And that's one hell of a thing. For Barry, the world has always been divided between the people who think they know what they'll be thinking in six months and those who don't.

And in his mind, his home computer has gotten him across that line and "That's one hell of a thing." For Barry, part of what it means to have crossed the line is to start to call the line into question. When he was in school, his inability to do the kind of mathematics he had "respect" for made him lose respect for himself as a learner, The computer put mathematics in a form that he could participate in. Barry has three children, has bought them their own calculators, and encourages them to "mess around with the computer."

He feels that they are going through the same problems with maths and science that he had and he wants them to have "a better start." For Barry, the computer holds the promise of a better start, not because it might teach his children a particular subject, but because it "might change their image of themselves. They might think of themselves as learners." Personal computers are certainly not the only hobby that people use to enhance their sense of identity. For my informants, "hobbies" have always been a way of life. Almost 90 per cent of them had been involved in a hobby other than computation, most usually in another "technical" hobby, such as photography, ham radio, or model railroading.

Fifteen percent of the hobbyists surveyed were using their computers to "augment" their participation in another hobby—for example, using the computer to keep an inventory of motorcycle parts, figure out ideal compression ratios for racing cars, interface with amateur radio equipment. For nearly a third of them, their home computer had completely replaced another hobby. People spoke of these abandoned hobbies as "fun" and as "good experiences," but their remarks about past hobbies underscored several ways in which in our day and time a computer hobby can be special. In particular, people spoke about their "switch to the computer" as making them part of something that was growing and that the society at large "really cared about."

Gregory is in his mid-forties, and has been in the electronics industry for all of his working life, as a technician, a programmer, and currently as a products designer. For two years, his computer shared space in his study with an elaborate model railroad system. A year and half before we met him he had bought a new hard copy printer and a graphics plotter. In the overcrowding that followed, the trains had finally found their way to storage in the basement.

We heard many echoes of Gregory's phrase, "it's part of the real world too." Hobbyists spoke about the computer offering them a connection with something beyond the hobby. For some, having a computer and "getting good at it" means crossing a frontier that separates "tinkering" from "real technology." They feel that the world sees their computer hobby as serious (several commented that friends and neighbours hardly even look at it as a hobby, as though the word were reserved for frivolities), and they start to see themselves that way too. Most first-generation hobbyists have technical educations, but many of them, like Barry, feel they have never been part of what is most exciting and important in the scientific and technical cultures. They see themselves as the low men on the totem pole.

Working with computers, even small computers, feels technologically "*avant garde*." A smaller group of hobbyists (but a group whose numbers are growing as new generations of personal computers become more accessible to the non-specialist) have always felt completely left out of the scientific and technical worlds. For them, owning a computer can mean crossing a "two cultures" divide. Alan, a 29-year-old high school French teacher who describes himself as "having a love affair with a TRS-80," has always felt he wasn't "smart enough to do science."

Alan majored in French ("It was easy for me ... my mother is from Montreal") and took up carpentry as a hobby. And although he was good at it, it only reinforced his sense of not being able to do intellectual things, which in his mind meant not being able to "do anything technical." When Barry began to do mathematics with his calculator, he felt that he started to cross a line to become the kind of person who could expect change and excitement in his intellectual life. For Alan, his TRS-80 led him across a line to become a member of a different culture, a scientific culture, a culture of "powerful people." For Barry, Alan, Gregory, relationships with computation enhanced self-image.

There is another way in which working with a computer can influence an individual's sense of identity. Ideas about computers, about how they work and what they can and cannot do, can be used to assert ideas about people. They can provide metaphors for thinking about oneself. Many hobbyists, fascinated by the idea of someday being able to trace out the complex relationships of electronic events, machine and assembly language instructions, and higher-level language commands within the computer, used the image of these many levels of intelligence to think about how people might work, about how people might, or might not, be like machines.

Some of this epistemology was implicit, for example when people made comments about "using the computer's software to think about my wetware." And sometimes, although less frequently, the issue became quite explicit. Conversations that began with descriptions of household "robotics" projects, such as a plan to build an energy monitoring system, led to reflections on how these projects would require programmes that could represent the system's knowledge to itself, and from there into formal epistemological reveries: were these the kind of self-representing programmes that ran inside people's heads? Do people have different kinds of self-representation programmes for representing different kinds of knowledge, such as the knowledge of a dream and the knowledge of being awake? What kind of self-representation programme might be running within people that allows them to remember and then forget their dreams?

OCCURRING REALITY

Hobbyist's descriptions of what it is like to work with their own computers frequently referred to the idea that the computer provides a safe corner of reality. Other hobbies can give a similar sense of security but often exact a price. For example, people can feel safe but limited. Alan, the French major, now "in love with his TRS-80," felt secure in his carpentry hobby, but he experienced it as a safety that came from refusing challenge. "It was an 'arty' hobby. We couldn't see myself any other way." The computer is more likely than most other media to allow the experience of playing worlds (let us call them "microworlds") that are secure and also adventurous enough to allow for mind-stretching explorations. Almost all of the hobbyists we interviewed described some version of a limited, safe, and transparent microworld that was embodied in their personal computer.

For Alan just the fact of working with a computer created such a world. For others, it was more specific: a morse code microworld; a text editor microworld; and, most generally, the assembly language microworld. This use of the computer as a place to build a microworld is particularly salient for children when they are put in computational environments in which they have access to programming. Elsewhere I shall report on mv study of elementary schoolchildren and adolescents who have gone through an experience with a LOGO computer system—a System designed to allow children to work in visually represented microworlds that are sufficiently constrained for a child to be able to understand and control them, yet sufficiently open-ended to give a child a real experience of intellectual power. Here I will mention only briefly one child and what she made of the computer as a medium for world building. Deborah at 11 years old was the baby of her family, the youngest of three children. Her childhood, had been spotted with illnesses which further exaggerated her "baby" position.

The members of her family were always doing things for her, insisting that she was not old enough to do the things that she wanted to do most: take out the laundry, baby-sit, stay over at a friend's house, choose her own hair style and her own clothes. Dependent on others at home, very overweight, and with an image of herself as sick and weak, Deborah had little sense of her own boundaries, her ability to say no, to assert control. Even at 11, she had become involved with a crowd of older kids who were smoking, drinking, using drugs. Towards the end of her eleventh year a LOGO computer came into Deborah's classroom as part of an educational experiment. At first, she found the computer frightening and threatening: until one day she hit upon the idea of confining the designs she made with the computer to ones in which the lines always came together in multiples of 30 degrees. She called it her "30 degrees world."

This restriction defined for her a space in which she felt safe but in which she was able to produce designs of great ingenuity and complexity. When I interviewed her two years later I found that she had used her experience with the "30 degrees world" as a kind of model, an experience-to-think-with. In her mind it represented how you could take control and make things happen by the judicious use of constraint. In Deborah's words, it was the first time that she ever "laid down her own laws." It was a turning point in her ability to take control of other situations. She lost 20 pounds, has given up smoking and drugs, and says that she "only sometimes" has a drink. For Deborah and for many adult hobbyists the sense of safety with the computer derived from the feeling of working in a sphere of intelligibility and transparency, a sphere that is protected, much as the space of a psychotherapeutic or psychoanalytic relationship is set off, bracketed.'

People talked about feeling safe and secure in the world they had built with their home computers, a world where there were few surprises and "things didn't change unless you wanted them to." Of course, there was much talk of problems, of false starts, of frustrations. There are "bugs" in hardware and in programmes. Things don't work; things go wrong. But bugs, with time, are either fixed or become "known" bugs. Joe is an insurance salesman in a small North California suburb who owns a second-hand Commodore Pet "with a lot of hardware problems." To Joe the bugs in his system "have become almost like friends": "We turn on the machine and we systematically check for my 'old friends,' and we swear, finding them there has a certain reassuring element."

FOLKS ON COMPUTER

No one can say exactly when the computer generation began-certainly not earlier than the 1960s, when computers began appearing in schools. But even computer whizzes in their twenties are acutely aware of how soon they are likely to be outstripped by today's grade schoolers. Says Steven Jobs, the multimillionaire co-founder of Apple Computer Inc.: "These kids know more about the new software than we do." New York computer executive Charles Lecht goes further: "If you were born before 1965, boy, you're going to be out of it." Where their parents fear to tread, the microkids plunge right in, no more worried about pushing incorrect buttons or making errors than adults are about dialling a wrong telephone number. Says mathematician Louis Robinson, IBM's resident computer sage: "They know what computers can and cannot do, while adults still regard them as omnipotent." Hughes Aircraft Chairman Allen Puckett, who used to share an Apple with son Jim, 12 said: "A lot of adults grew up in a-slide-rule world and still reject computers. But computers are as natural to kids as milk and cookies."

More and more members of the computer generation are tasting the heady pleasure of teaching their own teachers how to use the machines and, if they are lucky enough to have computers at home, instructing their parents as well. Says Ridgewood's Newman, a regular teacher of teachers: "It's a sort of mutual doorway. The barriers between adult and child, between teacher and student, are broken, and it's person to person. Nobody's looking down on anyone; they're looking each other right in the eye." Often adults find it easier to ask a child how to do something than to ask another adult. Says University of Kansas education professor Mary Kay Corbitt: "One adult student of mine brought her son to computer class, and we discovered that he was doing her assignment while she watched. Two weeks later she overcame her anxieties and was participating fully."

Confronted with the strange and unsettling world of the computer, teachers can get a useful perspective on what it is like to be a student again. After taking part in an elementary course in programming, Lois Brown, 54, a Wausau grade-school teacher, is thoroughly chastened. "Now we realize how little kids feel when there's a concept they don't understand. We sat in that course not wanting anyone to know all the things we didn't understand." Despite their obvious wariness of computers, parents are taking the lead in getting them into the schools. In Florida, communities have staged cake and candy sales, carnivals and tree plantings, weekend car washes, even a bike-athon to raise funds to buy computers. Says Marilyn Neff of Miami: "We feel computers will be the new paper and pencil." Of the 250 computers in the schools of Utica, Mich., more than two-thirds have been purchased by parent-sponsored fund drives.

Says Utica Principal Paul Yelinsky: "Moms and dads are coming in and telling the counsellors they have to get their kids in computer classes because it's the wave of the future." So important is computer literacy that the Alfred P. Sloan Foundation is beginning a major programme to get even such traditional liberal arts schools as St. John's College in Maryland to begin giving courses in it. Though many schools began purchasing computers with federal aid budget cutbacks are drying up that well. Apple's jobs points out that other nations, especially Britain, France and the Soviet Union though surprisingly not the electronics-minded Japanese are paying far more attention to computer education than is the US. Earlier this year, Jobs persuaded California Congressman Pete Stark and Missouri Senator John Danforth to introduce bills in Congress that would allow computer manufacturers to take a hefty tax write-off for any machines they donate to elementary and high schools. Under the present law, full deductions for such scientific equipment are allowed only if it is given to colleges and universities.

Steve Jobs originally spoke of giving an Apple to every public elementary and secondary school in the country, more than 80,000 computers worth as much as $200 million retail. He thought private schools should be included and encouraged other manufacturers to join in the programme as well. Meanwhile, Apple's archrival, the Tandy Corp., maker of the Radio Shack computer line, is taking a different tack: it has pledged $500,000 in equipment to spur development of educational programming, or courseware, for the classroom. Many of the approximately 100,000 computers now in US schools-roughly one for every 400 students-are in affluent suburbs like Ridgewood, a national leader in computer education. But the machines are also found in the unlikeliest of places.

On a Chippewa Indian reservation in Wisconsin, computers are being used by young members of the tribe to learn their ancient and nearly forgotten language. Alaska's small rural schools have been ordering computers to meet a special need:

> They allow students of different ages and abilities in the same small classrooms to learn at their own pace. Dubuque, Iowa, the New Yorker founding

> editor Harold Ross disdainfully located his provincial old lady, has 13 machines and another 20 on order. Bill Holloway, a professor of, computer education at the University of Kansas, calls the spread of small computers in the classroom nothing less than an avalanche. According to various industry studies, there may be from 300,000 to 650,000 computers in the schools by 1985.

So far, the most common, and least interesting, way to use school computers is in direct drill and review. The machine simply quizzes, prods and grades the student, very much like a robot teacher. Hundreds of programmes of this type are available for popular computers like the Apple II Plus, Radio Shack's TRS-80 and the Commodore PET. But many of these programmes are little more than computerized rehashes of the old classroom flash cards that go back to the days of McGuffey's readers. One notable difference: today when the student answers correctly, the screen will light up with WOWS, HOORAYS or smiling animals. Wrong answers may produce sad or scowling faces, perhaps accompanied by a falling tear. Partly because of teachers' fears of the machines and for their jobs and partly because of the poor quality of software, the frequently heralded electronic revolution in the classroom has been slow to occur. Now, however, it is being pushed along by steady improvements in teaching programmes, thanks to imaginative enterprises like the Minnesota Educational Computing Consortium.

One of its more refreshing drills: a programme called Wrong Note, which helps teach sight reading of musical scores. As a simple tune emanates from the computer's loudspeaker, matching notes appear on the screen, but sometimes the quiz intentionally errs and obliges students to find the false note. In order to do so, they can order up a repetition of the tune as often as Bogie and company did in Casablanca. Says Kenneth Brumbaugh, director of the consortium's instructional services:

> "Imagine asking a teacher to play it again and again!" Even very young children can profit from such exercises. At the Nordstom Elementary School in Morgan Hill, Calif., a suburb of San Jose, Colin Devenish, 7, is working with a classmate on the arithmetic drill, honing his skills in addition and subtraction. Unlike youngsters doing such drilling in the past, Colin seems to be enjoying himself enormously. Why? "Because," he replies mischievously within earshot of his teacher, "the computer doesn't yell."

Computers, operated only by touching a few buttons, are also remarkably effective devices for educating the handicapped. At the California School for the Deaf in Fremont, Rhonda Revera, 16, has worked with computers for five years, studying every subject from fractions to spelling. Rhonda offers a paean to the machine in sign language: "Computer makes me remember. It is fast, easy and better than writing on paper." Still another important use of computers is as a remedial tool. One is a spelling drill with a special incentive built into it: if all the answers are correct, a video game pops onto the screen as a reward. When one youngster worked his way through the drill, even classroom hecklers were impressed. Said one: "Hey, Old Wentworth's getting better." More entertaining and demanding are think tank-type strategy games like Geography Search, which launches competing teams on a Columbus-like voyage of exploration. They must make their way across the Atlantic, taking into account currents and winds, finding their longitude and latitude by means of star patterns and the length of a shadow thrown by a stick at high noon (methods that worked for Columbus, after all), and coping with such unforeseen perils as an outbreak of scurvy, an attack by pirates and a tropical storm. Only shrewd planning, wise choices and cooperative action ensure survival. The simulated voyage becomes uncannily real to the participants. Says the game's creator, Thomas Snyder, 31, who heads Computer Learning Connection, Inc., of Cambridge, Mass.: "When they get near the end and the computer finally shows them another ship near by, they act as if they had actually spotted a ship at sea."

Until a few years ago, the few computers available in secondary schools were essentially "dumb" terminals linked by telephone lines to a large, centrally located machine that served a variety of users through an arrangement called time-sharing. All the courseware was stored in the big computer's powerful memory, which could be tapped at will by students and teachers. The most successful example of such a system-and the one still used by Wisconsin's Chippewa Indians-is PLATO (for Programmed Logic for Automatic Teaching Operations).

Developed in the 1960s by the University of Illinois and Control Data Corp., PLATO is an exemplary teacher containing more than 8000 hours of courseware, much of it in a continuous curriculum. Thus, if a youngster forgets a point from an earlier lesson, PLATO will search its prodigious memory and patiently recapitulate. But such time-sharing schemes are extremely expensive, since they require open lines to the central computer. They also can become backed up at peak hours, and do not always lend themselves readily to what is the most intellectually demanding use of the computer: learning how to programme it. For this, the inexpensive, easy-to-operate personal computer, entirely self-contained and relying on equipment immediately at the student's side, is an ideal instrument-much more "user-friendly," as manufacturers like to say, than big machines. Yet even with a handy micro, programming can overwhelm the uninitiated. The programmer and computer must "speak" a common language. In the early days of the digital computer, this was, extremely difficult.

The machine reduces all the information it receives, whether it arrives as letters, numbers or graphic symbols, into the simplest possible electronic statements: either a yes or a no, represented by pulses of high or low voltage. To command the machine in its own internal language meant writing out endless strings of ones or zeroes, called bits and bytes, symbolizing those yes or no statements. But scientists soon began creating alternate languages for communicating with the machines that vaguely resemble everyday speech. The most popular of these computer tongues is BASIC (for Beginner's All-purpose Symbolic Instruction Code).

Developed at Dartmouth by mathematician John Kemeny and his colleague Thomas Kurtz to let even the least mathematically gifted student converse with the university's computers, it is "understood" by virtually all of today's personal computers. To show just how easy the language is, Kemeny offers this extremely simple lesson in programming: tell the computer to find the square roots (i.e., the numbers that, when multiplied by themselves, yield the original numbers) of eleven successive values, say 20 through 30.

The entire operation can be accomplished for a programme of just four steps:

```
1  FOR N= 20 TO 30
2  PRINT N, SQR(N)
3  NEXT N
4  END
```

Translated into everyday language, the first line tells the computer to let N stand successively for 20 through 30. The second instructs the machine to print the first value of N (that is, the number 20), compute its square root (SQR) and print out the result. The third tells the computer to go on to each of the succeeding values all the way through 30. Finally, the programme tells the computer to call it a day, its job having been done. Even the smallest machine can do such calculations in a flash, compared with the hours of work they might require of human computers. To preserve their creativity, the students can readily store their programmes on magnetic tape or on a small, 45 rpm-size plastic record called a floppy disc—which is not, as some parents believe, a new form of back injury. Then when the occasion arises for using the programme again, the computer operator merely loads the instructions back into the machine and punches in some new values for N.

The same broad principles apply to the creation of all software, even complex simulations like Geography Search. Literal-minded brutes that they are, computers do exactly what they are told. No more and no less. But youngsters of even the most tender age are surprising educators by showing they can master the beasts with startling ease. Computer software expert Leona Schauble of the Children's Television Workshop (producers of Sesame Street) recalls getting an eight-year-old boy at Manhattan's Little Red School House started on a simple computer game.

The game generated an image of a frog that would leap up and catch a butterfly, provided the right buttons were hit. After a few minutes, she checked back and found the frog jumping in slow motion. When she asked the youngster what happened, he replied, "Well, we wanted to make the frog catch more butterflies. So we got a listing of the variables and slowed him down." In other words, the youngster had broken into the game's programme and changed it to suit himself.

To instruct very young children, even Kemeny's BASIC is much too mathematical. Instead, more and more schools are turning to an innovative computer language called LOGO (from the Greek word for reason), developed by Seymour Papert and his colleagues at MIT. A mathematician who studied with the Swiss psychologist Jean Piaget, Papert has become something of a guru of the computer generation, predicting that the machines will revolutionize learning by taking much of the mystery out of mathematics, science and technology. Says he: "The computer can make the most abstract things concrete."

With a deceptively simple set of commands, LOGO enables youngsters who know nothing of geometry and algebra, and barely know how to read, to manipulate a triangular figure, dubbed the Turtle, on a computer screen and trace all manner of shapes with it. At the Lamplighter School in Dallas, teachers using LOGO get youngsters of three or four to write simple computer instructions. In one game, they manoeuvre "cars" and "garages" on the computer screen in such a way that the cars are parked inside the garages. While playing with LOGO, the youngsters learn simple words, the difference between left and right, and geometric concepts that they would not ordinarily encounter until junior high.

The machines crop up in the lives of youngsters even before they enter school-and sometimes before they learn to walk or talk-in the guise of such siliconised gadgetry as Little Professor and Speak & Spell. With a few presses of the button, these computerized games produce flashing lights, squealing sounds and disembodied voices that inculcate the rudiments of spelling and calculating. A record of sorts may have been set by Corey Schou, a computer scientist at the University of Central Florida in Orlando: he rigged up a home computer so his five-month-old daughter could operate it by pressing buttons in her crib and changing the designs on a nearby screen. Says the proud papa: "Basically, it's an electronic kaleidoscope, another diversion, another learning device." Whatever it is, it prepares youngsters for all those buttons they will encounter soon enough in and out of school. Parents and teachers may shudder at the thought, but it is only a short hop from skilful operation of a video game to learning fundamentals of programming. Says MIT Sociologist Sherry Turkle, 33, who has been studying the youthful computer culture for five years: "The line between game playing and programming is very thin. Programming takes what is powerful about games this articulation of knowledge, this learning about strategy and carries it to a higher level of power." By the time the youthful programmers reach the eighth or ninth grade, their skills may reach a marketable level. In Chicago, Jonathan Dubman, 14, and Kay Borzsony, 13, have formed a company called Aristotle Software to sell their own computer games and graphics programmes. Says Kay: "The nice thing about the computer business is that there is no real bias against children. In the computer magazines, you read articles by 12 and 13 year-olds." Laura Hyatt, 15, of Ridgewood, helps a stymied local insurance office figure out how to use its software. Says she: "It's better than babysitting." And, at $3.50 an hour, somewhat more profitable.

The prodigy of prodigies may be Eugene Volokh, 14, of Los Angeles. A Russian emigre, he earns $480 each week by doing 24 hours of programming for 20th Century-Fox, while carrying a full load of courses as a junior at UCLA. This year Greg Christensen, 18, of Anaheim, Calif., could make $100,000 in royalties from a video game he developed that was bought by Atari. Other youngsters are waiting at the sidelines in hopes of catching up with these young entrepreneurs. Every Tuesday night, Scott Whitfield, 13, and his brother Shawn, 11, appear at the Menlo Park, Calif., public library to get computer instruction. Says Scott: "We'll probably never get a job if we don't learn how to use a computer." Not all youngsters take equally to the machines. In a typical computer class, only about one in five students becomes seriously involved. Says Steven Scott, 16, of Wausau's West High: "Either you get the hang of it or you don't." Even so dedicated a computernik as Ridgewood's Nick Newman finds programming interesting only for a purpose. His own goal is to apply his computer knowledge to a career in science or medicine. Whatever these youngsters make of their computer experiences, they will surely confront the world differently from their parents.

The precise, orderly steps of logic required to use and programme the machines promise to shape-and sharpen-the thought processes of the computer generation. Indeed, the youngsters playing all those strategy games are doing precisely what corporations do when they plan to launch a new product, or what military leaders do when they devise strategies to confront a potential foe.

Whether such abilities will change the world for the better is another matter. Princeton psychologist George Miller, for one, has doubts that "a few years of thinking like a computer can change patterns of irrational thought that have persisted throughout recorded history."

Other social critics ask if clear thinking is enough - if, in fact, there might not be a danger in raising a generation to believe that it has the analytical tools to contemplate any problem. Says MIT computer science professor Joseph Weizenbaum: "There's a whole world of real problems, of human problems, which is essentially being ignored." It is still impossible, after all, to reduce a human relationship to a printout or to solve a moral question by bits and bytes.

Some critics predict a future not unlike that portrayed in Isaac Asimov's I, *Robot,* a science fiction novel set in a society so thoroughly computer-dominated that the people cannot do arithmetic. Humanist critic George Steiner acerbically calls the computer generation the advance guard of a breed of "computer-mutants." Says Steiner: "They will be out of touch with certain springs of human identity and creativity, which belong to the full use of language rather than mathematical and symbolical codes."

Many others are much more sanguine. University of Chicago philosopher of science Stephen Toulmin predicts that computers will "re-intellectualize" the television generation. "TV relieved people of the necessity to do anything," says Toulmin. "Computers depend on what you do yourself." Catholic theologian David Tracy argues that "using computers sharpens the mind's ability to deal with our world: the world of technology." The final word may be simpler, and not pronounced by elders who find a devilish soul in the new machine. More so than adults, the young know the computer for what it is. Says a 10-year-old at Manhattan's Bank Street School: "It's dumb. We have to tell it everything." They also know something important is afoot. Says Shawn Whitfield: "When I grow up it's going to be the Computer Age.

VARIOUS APPROACHES

The use of the computer as a medium for building a transparent and intelligible world brings us to how it can be used to think through questions of political ideology. Fred sells components for a large electronics supply house. He narrowly escaped starvation in a prisoner of war camp during World War II, and from that experience he says that he took " a sense of optimism." "we mean, if there is something out there and you want to do it-do it, understand it, act." Fred has tried to live that way. He is active in local politics; he keeps up with the news; he writes letters to the editor of his town newspaper. He bought his TRS80 on an impulse because "it seemed that you wouldn't be able to understand American society any more if you didn't know about computers." When it comes to working with his computer, Fred wants to know "exactly how things work":

> "There is a big gap in my own mind between the fact that an electrical circuit can be on or off and the binary number system ... and again from there to the BASIC language. I've got to understand all of that."

When hobbyists like Fred spoke about "wanting to know exactly how things work" in their computers, they were usually talking about wanting to know how their systems were built up from level to level, from electrical circuit to high-level language command. Fred, for example, expressed sharp frustration at gaps in his ability to follow the system through:

"we can't really follow the continuum. I'm going to a user's group meeting, I'm talking to people and reading books and some of it is helping, but I am really frustrated. We want to be able to follow the whole thing through."

Larry, 35, lives in a Boston suburb. His computer offered him the first way he ever found to challenge the school's judgments of his child's abilities. A year before we met him, Larry had bought an Apple computer for small business use and ended up bringing it home so that his children would be able to play with it. His 12-year-old son Todd had been judged "backward" by his teachers through six years of schooling. His maths scores were low; he could barely read. But Todd picked up the Apple manual, taught himself how to use the game packages, and then taught himself how to programme in BASIC. As far as anyone knows, the Apple manual was the first book that Todd had ever read. In three weeks Todd was writing his own games, games that demanded an understanding of variables and a knowledge of geometry that his teachers claimed he didn't have. Larry is starting to demand more from his son's teachers. His experience with Todd has made him optimistic about what computers will mean for politics because "people will get used to understanding things, of being in control of things, and then, will demand more." This optimism is widely shared among hobbyists, part of a distinct style of talking about computers and politics.

Hobbyists, like Fred and Larry, take what is most characteristic about their relationships with the computer—using computers to build safe microworlds of transparent understanding-and turn it into a political metaphor. Hobbyists associate images of computational transparency and of "knowing how the in machine works" with a kind of politics where relations of power will be transparent, where people will control their destinies, where work will facilitate a rich and balanced cognitive life, and where decentralized power will follow from decentralized information resources. For many hobbyists a relationship with their home computer carries longings for a better and simpler life in a more transparent society. *Co-Evolution Quarterly*, *Mother Earth News*, *Runner's World*, and *Byte Magazine* lie together on hobbyists' coffee tables.

Small computers become the focus of hopes of building cottage industries that will allow people to work out of their homes, have more personal autonomy, not have to punch time cards, and be able to spend more time with their family and out of doors. Some see personal computers as a next step in the ecology movement: decentralized technology will mean less waste. Some see personal computers as a way for individuals to assert greater control over their children's education, believing that computerized curricula will soon offer children better education at home than can be offered in today's schools. Some see personal computers as a path to a new populism:

> Personal computer networks will allow citizens to band together to send mail, run decentralized schools, information resources, and local governments. In sum, many of the computer hobbyists I have interviewed talk about the computers in their living-rooms as windows onto a future where relationships with technology will be more direct, where people will understand how things work, and where dependence on big government, big corporations, and big machines will end. They imagine the politics of this computer-rich future by generalizing from their special relationship to the technology, a relationship characterized by simplicity and a sense of control.

DECLINING FROM WORK

Over 40 per cent of those who responded to my survey worked, or had once worked, as computer programmers. The programmer is typically in a situation where he or she is in touch with only a very small part of the problem that is being worked on. Increasingly, programmers work in large teams where each individual has very little sense of the whole, of how it all fits together.

Programmers have watched their opportunities to exercise their skill as a whole activity being taken away (for those who are too young, the story of the process remains alive in the collective mythology of the shop). They have watched their work being routinized, being parcelled out into the well defined modules that make up the tasks of the structured programming team. They mythologize a golden age. This live experience at work made programmers particularly sensitive to the parcellization of knowledge and to the alienation from a sense of wholeness in work. And they bring this sensitivity to their home computer hobbies.

Hannah worked as a programming consultant for a large business system for ten years before starting her own consulting company through which she offers her services to other computer hobbyists. To her, nothing is more depressing than working on a tiny piece of a problem. In her old job, "most of the time we didn't even know what the whole problem was." She likes working with computers at home because she has more control of her time and can spend more time with her family. But she says that what is most important about working with a personal computer is that "we can finally think about a whole problem." Hannah's feelings were widely shared among the hobbyists interviewed, most notably among programmers, ex-programmers, and "team engineers." Images of lack of intellectual balance, of fragmentation, of not being connected, came up often, with the computer at home usually placed in the role of righting what had been wrong at work. As Hannah put it: "With my computer at home we do everything, we see my whole self, all my kinds of thinking."

For these people, having a computer at home meant "thinking experiences" where they could see what their real capacities were, where they had a chance to try things out. Karl, for example, who worked for a long time as a programmer and who is now an engineer in a microprocessor firm, had thought a lot about his mental "ecology."

In the first half of an engineering project there is literally nothing coming together and that's when we find that we need to go home and put something together. We used to make lamps like those in the living room out of glass bottles. But then towards the middle of an engineering project, things did start to come together and we would lose the need for cutting glass. But if you never or rarely get to finish things at work, if your job is basically to make little pieces and it's somebody else's job to make them fit into a whole, then working with the computer at home can give you an experience of getting it all together. You do the whole thing—building up from machine code to finished project. It makes you feel in balance.

When we spoke with Karl he was at a point where everything at work seemed "pretty disconnected." Or, as Karl put it, if he hadn't had his home computer, it would have been "glass cutting time". Karl saw his current work with the computer as a corrective to fragmentation on the job. During our interview, he spoke to me about his current plans for revamping his computer system. Computer systems like his which have several components (keyboard, display screen, printer) have to be connected by using devices, usually simple circuits, called "interfaces." Karl's plan called for using a separate microprocessor as an interface for each component. He had conflicts about the "rationality" of his project.

After all, he admitted, it was far from economical if one measures economy in terms of the cost of the hardware. Each interface circuit needed only a few specialized and inexpensive chips. There was no need for a general-purpose microprocessor at each node. Specialized chips could do the job more cheaply, but could not satisfy his desire to experience the system as maximally coherent. By replacing the special-purpose circuits by separate, but identical, general-purpose microprocessors, the whole system, at least in Karl's eyes, became uniform, intelligible, and systematic. Karl could not help concluding: "We guess you could say that my choice of projects is not always rational." But from the perspective of using a relationship with computation as a way of "working through" personal concerns, Karl's rationale was clear. For Karl the "inexpensive" solution., using a collection of opaque, ad hoc circuits, felt unintelligible. It felt to him like his work situation. His plans for his multiprocessor system were dictated by the logic of compensation rather than by the logic of material economy.

BASICS OF MACHINES

In studying the hobbyist experience we have found people, largely people with technical backgrounds, in intense involvement with machines. They describe their work (or rather their leisure) with the computer as different from what they have done before with other hobbies. They describe it as an involvement with greater personal consequence. Some of the sense of consequence comes from an historical moment: the computer hobby is seen as signifying a place in the "*avant garde*". Although in some circles "computer person" is a term of derision, the hobbyist experiences it with pride.

Some of the sense of consequence comes from experiencing an individualistic and independent relationship with computation that can be mythologized as belonging to a now-past "golden age" of the programmer. But most of the sense of consequence comes from the holding power and intensity of the time spent with the computer. What is there about these people and these machines that makes possible relationships of such power and such intensity? For me, the relationships that hobbyists form with their home computers can be partially captured with a metaphor of the "mind" and the "body" of the machine.

The "mind" of the computer is that side of computation that involves thinking in terms of high-level programmes. In this metaphor, relating to the "body" of the computer means not only working on hardware, but also, and indeed especially, working with programmes in a way that is as close as possible to the machine code—that is to say, as close as possible to the core of the computer, its central processing unit (CPU). In terms of this metaphor I have found that the prototypical hobbyist is trying to get into a relationship with the body (rather than the mind) of the machine, to assert power and control in the relationship with the computer, and to create safe worlds of transparent understanding. In trying to find concepts for thinking more clearly about what draws the hobbyist to this kind of relationship with the CPU and about what its meaning might be, we find three issues particularly salient. We think, moreover, that, although we formulate them here in terms of computers, they are relevant to understanding relationships with other technologies as well. The first issue goes back to control.

The hobbyist complains of a work situation where everyone suffers from the constant presence of intermediaries. Bureaucracies stand between the programmer and the computer, a bureaucracy that schedules the computer, that decides its up and down time, that apportions the work for its software design and decides on priorities and procedures for access to it. At work, when something goes wrong with the system it is usually the fault of an intermediary person, one of the many "somebody elses" who deal with the machine. Or it may be the fault of a technical intermediary, one of the many elements in the computer system that mediate between the user and the bare machine: a compiler, an interpreter, an operating system, someone else's programme. At home, the hobbyists feel themselves as working directly with the CPU, in complete and direct control of the machine's power.

And when something does blow up, the situation has a special immediacy. It is between them and the bare machine. When a FORTRAN programme is run on a large IBM machine the events in the machine are far from being in one-to-one correspondence with the steps of code written by the programmer. Two factors contribute. First, it is in the nature of higher-level languages to work in a conceptual space different from that of the machine. FORTRAN works in a "formula" space, COBOL works in a "business" space, both very different from the space of bits and bytes. Second, the sense of indirect relationship is exacerbated when the compiled code is run by an operating system which allocates memory, mediates control of peripheral devices, and even interleaves the programme with other programmes. At home hobbyists can follow machine-language programmes step by step as their instructions pass through the CPU. They can envision the changes in state of the whole system as being produced by specific actions of the CPU.

And if they suspect that there is a bug in the hardware they can pull out an oscilloscope and see whether the CPU is doing what it should in response to a given instruction. They can figure out where the signals should be going, they can collect their own evidence for what is going wrong, trap and fix the bug themselves. Again and again in my interviewing I heard about the pleasures of debugging-of "going in with metres and scopes and tracking it down." The procedure exhilarates. With every successfully tracked bug comes an affirmation of direct control over the machine. The issue of control was often explicitly recognized by the hobbyists we interviewed. But they lacked a language for naming a second issue which has to do with a notion referred to as "syntonicity" within the psychoanalytic tradition.

Syntonicity implies that we should look for "body-to-body" identification in every powerful relationship with a technology—the body of the person and the body of the machine. It implies that we should understand the appeal of machine language in terms of people's ability to identify with what is happening inside the machine. The CPU of the hobbyist computer lends itself to personal identification with its primary action: moving something that is conceptually almost a physical object (a byte of information) in and out of some thing (a register) that is almost a physical place. The metaphor is concrete and spatial. One can imagine finding the bytes, feeling them, doing something very simple to them, and passing them on. For many of the people that we met in the hobbyist culture, getting into this kind of identification feels safe. It makes the machine feel real. There is a third issue raised by the hobbyists' relationship to the CPU. It is an aesthetic one.

The generation of hobby computers that was born in the 1970s are very primitive machines. The hobbyist thinks of much about them as "klugey," a computerist's way of saying that one is dealing with a compromise, a collection of patches whose structure has been dictated by arbitrary corporate decisions, by economic necessities. The corner of the hobbyist machine that seems to them to have the greatest "intellectual integrity," that distills what they feel to be a tradition of some of the best ideas in computer science, that comes closest to being "clean," is the CPU. And so it is natural for the hobbyist to seek the closest possible contact with it. For a culture in which there is a widely shared aesthetic of simplicity, intelligibility, control, and transparency, getting into the "un-klugey" part of the machine and working in machine code seems the most aesthetically satisfying way to use the personal computer as an artistic medium.

CHILDREN AT COMPUTER

Instrumental uses of the computer to help people think have been dramatized in science fiction. Many cultural barriers impede children from making scientific knowledge their own. Among these barriers the most visible are the physically brutal effects of deprivation and isolation. Other barriers are more political. Many children who grow up in our cities are surrounded by the artifacts of science but have good reason to see them as belonging to "the others"; in many cases they are perceived as belonging to the social enemy. Still other obstacles are more abstract, though ultimately of the same nature.

Most branches of the most sophisticated modern culture of Europe and the US are so deeply "mathophobic" that many privileged children are as effectively (if more gently) kept from appropriating science as their own. Space-age objects, in the form of small computers, will cross these cultural barriers to enter the private worlds of children everywhere. They will do so not as mere physical objects. This chapter is about how computers can be carriers of powerful ideas and of the seeds of cultural change, how they can help people form new relationships with knowledge that cut across the traditional lines separating humanities from sciences and knowledge of the self from both of these. It is about using computers to challenge current beliefs about who can understand what and at what age. It is about using computers to question standard assumptions in developmental psychology and in the psychology of aptitudes and attitudes. It is about whether personal computers and the cultures in which they are used will continue to be the creatures of "engineers' alone or whether we can construct intellectual environments in which people who today think of themselves as "humanists" will feel part of, not alienated from, the process of constructing computational cultures.

But there is a world of difference between what computers can do and what society will choose to do with them. Society has many ways to resist fundamental and threatening change. Thus, this book is about facing choices that are ultimately political. It looks at some of the forces of change and of reaction to those forces that are called into play as the computer presence begins to enter the politically charged world of education.

In many schools today, the phrase "computer-aided instruction" means making the computer teach the child. One might say the computer is being used to programme the child. The child programmes the computer and, in doing so, both acquires a sense of mastery over a piece of the most modern and powerful technology and establishes an intimate contact with some of the deepest ideas from science, from mathematics, and from the art of intellectual model-building.

The learning paths have led hundreds of children to becoming quite sophisticated programmers. Once programming is seen in the proper perspective, there is nothing very surprising about the fact that this should happen. Programming a computer means nothing more or less than communicating to it in a language that it and the human user can both "understand." And learning languages is one of the things children do best. Every normal child learns to talk. Why then should a child not learn to "talk" to a computer? There are many reasons why someone might expect it to be difficult. For example, although babies learn to speak their native language with spectacular ease, most children have great difficulty learning foreign languages in schools and, indeed, often learn the written version of their own language none too successfully. Isn't learning a computer language more like the difficult process of learning a foreign written language than the easy one of learning to speak one's own language? And isn't the problem further compounded by all the difficulties most people encounter learning mathematics?

Two fundamental ideas run through this chapter. The first is that it is possible to design computers so that learning to communicate with them can be a natural process, more like learning French by living in France than like trying to learn it through the unnatural process of American foreign-language instruction in classrooms. Second, learning to communicate with a computer may change the way other learning takes place. The computer can be a mathematics-speaking and an alphabetic speaking entity. We are learning how to make computers with which children love to communicate. When this communication occurs, children learn mathematics as a living language.

Moreover, mathematical communication and alphabetic communication are thereby both transformed from the alien and therefore difficult things they are for most children into natural and therefore easy ones. The idea of "talking mathematics" to a computer can be generalized to a view of learning mathematics in "Mathland"; that is to say, in a context which is to learning mathematics what living in France is to learning French.

It is generally assumed that children cannot learn formal geometry until well into their school years and that most cannot learn it too well even then. But we can quickly see that these assumptions are based on extremely weak evidence by asking analogous questions about the ability of children to learn French. If we had to base our opinions on observation of how poorly children learned French in American schools, we would have to conclude that most people were incapable of mastering it. But we know that all normal children would learn it very easily if they lived in France. Much of what we now see as too "formal" or "too mathematical" will be learned just as easily when children grow up in the computer rich world of the very near future. We use the examination of our relationship with mathematics as a thematic example of how technological and social processes interact in the construction of ideas about human capacities. And mathematical examples will also help to describe a theory of how learning works and of how it goes wrong. Take from Jean Piaget a model of children as builders of their own intellectual structures. Children seem to be innately gifted learners, acquiring long before they go to school a vast quantity of knowledge by a process call "Piagetian learning", or "learning without being taught." For example, children learn to speak, learn the intuitive geometry needed to get around in space, and learn enough of logic and rhetorics to get around parents-all this without being "taught". We must ask why some learning takes place so early and spontaneously while some is delayed many years or does not happen at all without deliberately imposed formal instruction.

MADE FOR CHILDREN

If we really look at the "child as builder" we are on our way to an answer. All builders need materials to build with. Where we are at variance with Piaget is in the role we attribute to the surrounding cultures as a source of these materials. In some cases the culture supplies them in abundance, thus facilitating constructive Piagetian learning. For example, the fact that so many important things (knives and forks, mothers and fathers, shoes and socks) come in pairs is a "material" for the construction of an intuitive sense of number.

But in many cases where Piaget would explain the slower development of a particular concept by its greater complexity or formality, we see the critical factor as the relative poverty of the culture in those materials that would make the concept simple and concrete. In yet other cases the culture may provide materials but block their use. In the case of formal mathematics, there is both a shortage of' formal materials and a cultural block. The mathophobia endemic in contemporary culture blocks many people from learning anything they recognize as "math," although they may have no trouble with mathematical knowledge they do not perceive as such.

We shall see again and again that the consequences of mathophobia go far beyond obstructing the learning of mathematics and science. They interact with other endemic "cultural toxins," for example, with popular theories of aptitudes, to contaminate people's images of themselves as learners. Difficulty with school math is often the first step of an invasive intellectual process that leads us all to define ourselves as bundles of aptitudes and inaptitudes, as being "mathematical" or "not mathematical," "artistic" or "not artistic," "musical" or "not musical," "profound" or "superficial," "intelligent" or "dumb."

Deficiency becomes identity and learning is transformed from the early child's free exploration of the world to a chore beset by insecurities and self-imposed restrictions. Two major themes—that children can learn to use computers in a masterful way, and that learning to use computers can change the way they learn everything else—have shaped research agenda on computers and education. The metaphor of imitating the way the child learns to talk has been constantly with us in this work and has led to a vision of education and of education research very different from the traditional ones. For people in the teaching professions, the word "education" tends to evoke "teaching," particularly class-room teaching. The goal of education research tends therefore to be focused on how to improve classroom teaching. But if, the model of successful learning is the way a child learns to talk, a process that takes place without deliberate and organized teaching, the goal set is very different.

The classroom as an artificial and inefficient learning environment that society has been forced to invent because its informal environments fail in certain essential learning domains, such as writing or grammar or school math. The computer presence will enable us to so modify the learning environment outside the classrooms that much if not all the knowledge that schools presently try to teach with such pain and expense and such limited success will be learned, as the child learns to talk, painlessly, successfully and without organized instruction. This obviously implies that schools as we know them today will have no place in the future. But it is an open question whether they will adapt by transforming themselves into something new or will wither away and be replaced.

Although technology will play an essential role in the realization of our vision of the future of education, our central focus is not on the machine but on the mind, and particularly on the way in which intellectual movements and cultures define themselves and grow. Indeed, the role we give to the computer is that of a carrier of cultural "germs" or "seeds" whose intellectual products will not need technological support once they take root in an actively growing mind. Many if not all the children who grow up with a love and aptitude for mathematics owe this feeling, at least in part, to the fact that they happened to acquire "germs" of the "math culture" from adults, who, one might say, knew how to speak mathematics, even if only in the way that Moliere had M. Jourdain speak prose without knowing it. These "math-speaking" adults do not necessarily know how to solve equations; rather, they are marked by a turn of mind that shows up in the logic of their arguments and in the fact that for them to play is often to play with such things as puzzles, puns, and paradoxes. Those children who prove recalcitrant to math and science education include many whose environments happened to be relatively poor in math-speaking adults. Such children come to school lacking elements necessary for the easy learning of school math.

School has been unable to supply these missing elements, and, by forcing the children into learning situations doomed in advance, it generates powerful negative feelings about mathematics and perhaps about learning in general. Thus, is set up a vicious self-perpetuating cycle. For these same children will one day be parents and will not only fail to pass on mathematical germs but will almost certainly infect their children with the opposing and intellectually destructive germs of mathophobia. Fortunately, it is sufficient to break the self-perpetuating cycle at one point for it to remain broken forever.

The Turtle is a computer-controlled cybernetic animal. It exists within the cognitive minicultures of the "LOGO environment," LOGO being the computer language in which communication with the Turtle takes place. The Turtle serves no other purpose than of being good to programme and good to think with. Some Turtles are abstract objects that live on computer screens. Others, like the floor Turtles shown in the frontispiece, are physical objects that can be picked up like any mechanical toy. A first encounter often begins by showing the child how a Turtle can be made to move by typing commands at a keyboard. FORWARD 100 makes the Turtle move in a straight line a distance of 100 Turtle steps of about a millimetre each. Typing RIGHT 90 causes the Turtle to pivot in place through 90 degrees. Typing PENDOWN causes the Turtle to lower a pen so as to leave a visible trace of its path, while PENUP instructs it to raise the pen. Of course, the child needs to explore a great deal before gaining mastery of what the numbers mean. But the task is engaging enough to carry most children through this learning process.

The idea of programming is introduced through the metaphor of teaching the Turtle a new word. This is simply done, and children often begin their programming experience by programming the Turtle to respond to new commands invented by the child such as SQUARE or TRIANGLE or SQ or TRI or whatever the child wishes, by drawing the appropriate shapes. New commands, once defined, can be used to define others. For example, just as the house is built out of a triangle and a square, the programme for drawing it is built out of the commands for drawing a square and a triangle.

There are four steps in the evolution of this programme. From these simple drawings the young programmer can go on in many different directions. Some work on more complex drawings, either figural or abstract. Some abandon the use of the Turtle as a drawing instrument and learn to use its touch sensors to programme it to seek out or avoid objects. Later children learn that the computer can be programmed to make music as well as move Turtles and combine the two activities by programming Turtles to dance. Or they can move on from floor Turtles to "screen Turtles," which they programme to draw moving pictures in bright colours. The examples are infinitely varied, but in each the child is learning how to exercise control over an exceptionally rich and sophisticated "micro-world".

Readers who have never seen an interactive computer display might find it hard to imagine where this can lead. As a mental exercise they might like to imagine an electronic sketchpad, a computer graphics display of the not-too-distant future. This is a television screen that can display moving pictures in colour. You can also "draw" on it, giving it instructions, perhaps by typing, perhaps by speaking, or perhaps by pointing with a wand.

On request, a palette of colours could appear on the screen. You can choose a colour by pointing at it with the wand. Until you change your choice, the wand draws in that colour. Up to this point the distinction from traditional art materials may seem slight, but the distinction becomes very real when you begin to think about editing the drawing. You can "talk to your drawing" in computer language. You can "tell" it to replace this colour with that. Or set a drawing in motion. Or make two copies and set them in counter-rotating motion. Or replace the colour palette with a sound palette and "draw" a piece of music.

You can file your work in computer memory and retrieve it at your pleasure, or have it delivered into the memory of any of the many millions of other computers linked to the central communication network for the pleasure of your friends. That all this would be fun needs no argument. But it is more than fun. Very powerful kinds of learning are taking place. Children working with an electronic sketchpad are learning a language for talking about shapes and fluxes of shapes, about velocities and rates of change, about processes and procedures. They are learning to speak mathematics, and are acquiring a new image of themselves as mathematicians.

Some of the children were highly successful in school, some were diagnosed as emotionally or cognitively disabled. Some of the children were so severely afflicted by cerebral palsy that they had never purposefully manipulated physical objects. Some of them had expressed their talents in "mathematical" forms, some in "verbal" forms, and some in artistically "visual" or in "musical" forms. Of course, these children did not achieve a fluency in programming that came close to matching their use of spoken language. If we take the Mathland metaphor seriously, their computer experience was more like learning French by spending a week or two on vacation in France than like living there. But like children who have spent a vacation with foreign-speaking cousins, they were clearly on their way to "speaking computer."

When we have thought about what these studies mean we are left with two clear impressions. First, that all children will, under the right conditions, acquire a proficiency with programming that will make it one of their more advanced intellectual accomplishments. Second, that the "right conditions" are very different from the kind of access to computers that is now becoming established as the norm in schools.

The conditions necessary for the kind of relationships with a computer that we will be writing about in this book require more and freer access to the computer than educational planners currently anticipate. And they require a kind of computer language and a learning environment around that language very different from those the schools are now providing. They even require a kind of computer rather different from those that the schools are currently buying. It will take most of this chapter to convey some sense of the choices among computers, computer languages, and, more generally, among computer cultures, that influence how well children will learn from working with computation and what benefits they will get from doing so. But the question of the economic feasibility of free access to computers for every child can be dealt with immediately.

Our vision of a new kind of learning environment demands free contact between children and computers. This could happen because the child's family buys one or a child's friends have one. For purposes of discussion here (and to extend our discussion to all social groups) let us assume that it happens because schools give every one of their students his or her own powerful personal computer. Most "practical" people (including parents, teachers, school principals, and foundation administrators) react to this idea in much the same way: "Even if computers could have all the effects, you talk about, it would still be impossible to put your ideas into action. Where would the money come from?" What these people are saying needs to be faced squarely. They are wrong. Let's consider the cohort of children who will enter kindergarten in the coming year, the "Class of 2010," and let's do some arithmetic. The direct public cost of schooling a child for thirteen years, from kindergarten through twelfth grade, is over $20,000 today (and for the class of 2000, it may be closer to $30,000).

A conservatively high estimate of the cost of supplying each of these children with a personal computer with enough power for it to serve the kinds of educational ends described in this book, and of upgrading, repairing, and replacing it when necessary would be about $1000 per student, distributed over 13 years in school. Thus, "computer costs" for the class of 2000 would represent only about 5 per cent of the total public expenditure on education, and this would be the case even if nothing else in the structure of educational costs changed because of the computer presence.

But in fact computers in education stand a good chance of making other aspects of education cheaper. Schools might be able to reduce their cycle from 13 years to 12 years; they might be able to take advantage of the greater autonomy the computer gives the students and increase the size of classes by one or two students without decreasing the personal attention each student is given. Either of these two moves would "recuperate" the computer cost. Our goal is not educational economies: it is not to use computation to save a year off the time a child spends in an otherwise unchanged school or to push an extra child into an elementary school classroom.

GENERAL EFFECTS

George Mamunes, 14, a gangling ninth-grader dressed in flannel shirt, blue jeans and hiking boots, knits his thick, dark eyebrows while putting the finishing touches on a computer programme, already nearly 300 lines long. For those uninitiated in the special languages of the computer age, it looks like a hopeless mess of numerical gibberish. But when completed, these instructions should produce a computer image of the heart detailed enough to show every major artery and vein, as well as valves and chambers.

The electronic heart is part of a teaching tool George is putting together for eighth-grade biology classes. A few feet away sits Pam Miller, 14, a ninth-grader with long, brown hair draped far down her back. She is operating a computer programme or software that simulates the workings of a nuclear reactor. Today she is fine-tuning the section that governs the control rods, those regulators of the reactor's nuclear fires. Tapping away at the keyboard, Pam explains: "You have to maximize the power output without destroying the reactor." Suddenly, flashing numbers burst upon the screen. "There," says Pam, her face lighting up. "Reactor overheated. Power output low. Reactor core damaged. Melt-down!" A disaster that she has brought on intentionally, just to show how it could happen.

Other disciples, seated at terminals scattered around the room, are no less absorbed. Meilin Wong, 15, chic in blue velour blouse, jeans and Bass moccasins, is trying to figure out what went wrong with her business data management programme. She is an old hand at such troubleshooting, having spent much of last semester "debugging" a programme that, when printed-out, stretches over 30 ft. Jim McGuire, 13, is creating a video game called Spaceship, which will let electronic star warriors zap a boxy- looking orbital intruder.

A more mundane programme is emerging from 15 year old Dave McCann's terminal: a verb test for seventh and eighth-grade Spanish classes. Off in a corner two youngsters are putting the impish face of *Mad* magazine's cartoon hero, Alfred E. Neuman, onto the computer screen. Says Muller, as he presides proudly over these after-hours computer converts: "No one told them they have to be here. They're not usually doing assignments. They're experimenting. They're letting their imaginations run free." Muller's disciples are not all math whizzes. Or straight - A students. Or particularly precocious. They are reasonably normal youngsters who have grown up with computers. For them, in ways that few people over 30 can understand, manipulating these complex machines is as natural as riding a bike, playing baseball or even solving Rubik's cube. Like thousands of others across the country, they are part of a revolutionary vanguard: the computer generation. Not only is this generation propelling traditional education down promising avenues, it is tugging at the entire social fabric, foreshadowing changes at least as startling and momentous as those ushered in by a new generation of automobile users more than a half-century ago.

In the classroom, where youngsters are being introduced to the machines as early as kindergarten, they astound-and often outpace-their teachers with their computer skills. After school they gather at the mushrooming number of computer stores (more than 1500 at last count) or join the computer clubs that are becoming a regular part of the youthful landscape. Huddling around any available machine, they argue over their programmes, solve computer problems and swap software as intensely as kids once haggled over baseball cards.

In the summer, they may even go off to computer camps, another growth industry, and if they are Boy Scouts, they may try for a computer merit badge. During mischievous moments, they may tinker with one another's programmes, writing in steps that will flash an unexpected insult or obscenity across a buddy's video screen. Some try to pick the encoded electronic locks on copyrighted software, taking glee in outwitting their elders, or spin fanciful plots to break into computer networks.

A few turn their skills to profit by showing baffled businessmen how to get idle, new computers to run, or by establishing Delaware based corporations to market their own software creations. To the bafflement of their parents, they talk in a jargon of their own ("Hey, Charlie, you should have POKED instead of PEEKED"). As with so many other changes in contemporary life, the spark for this revolution is technological: a bit of silicon sophistication variously known as the personal, home or microcomputer. No larger than an attache case, apart from its video screen, this mighty mite packs the computing power of machines that two decades ago occupied a full room. Yet the microsi as they are affectionately called, are a relative bargain to buy and are becoming steadily cheaper.

Many models cost under $1000, bringing them within reach of schools, parents or the children themselves. Last week, in the sharpest price break yet, Timex announced it will begin selling a small home computer for a suggested retail price of $99.95. But size and price cannot explain why computers have taken such a strong hold on so many youngsters. Certainly their interest has been stirred by a related rage, video games, whose computer- generated flashes, zaps, and pings have not only all the appeal pinball machines had for their elders but go a significant step further: they pique young minds to learn more about all that electronic prestidigitation. But many experts, and most of the young operatives, agree that the overwhelming attraction of the machines is the lure of control, the pleasure of being able to think out and then make something happen, a satisfaction all too often denied children.

Recognized by students and teachers, alike as his school's best computer programmer, Lewis works afternoons as an instructor for a computer consulting firm, introducing younger children to the machines. Last year his employers sent him to Chicago, where he displayed his special teaching gifts before a meeting of educators. As Lewis told *Time* correspondent Peter Stoler, "we love these machines. I've got all this power at my fingertips. Without computers, we don't know what do be. With them, I'm somebody."

Perhaps because of the faintly macho side of computers, the bug seems to strike many more boys than girls in the pre-adolescent years. Says Steve Siegelbaum, Lewis's teacher: "Maybe it's because boys are pushed more towards math and logic than girls are. Maybe it's because boys are just more aggressive." Paradoxically, the computer passion is often stirred in youngsters who seem least likely to be interested in high tech. Jay Harstad, 12, of Minnetonka, Minn., litters his house with poems and sketches but will do almost anything to avoid doing his math homework. Yet Jay is one of the Gatewood Elementary School's premier computerniks and regularly helps teachers introduce fourth-graders to the machines. At West High School in Wausau, Wis., Chris Schumann, 16, a junior, has made a name for himself by translating musical notes into digital form and getting a computer to play Bach and Vivaldi through its loudspeaker. Originally, Chris regarded computers as remote and forbidding, but that changed when he was introduced to his first micro. "It looked real friendly," he says. "It didn't overpower you. It wasn't this ominous thing but something you could get close to."

The closeness can be contagious. Explains Nick Newman, 15, Muller's chief disciple at Ridgewood: "The more you do on the machine, the more enjoyable it gets. It becomes habit-forming." In Alpena, Mich., youngsters who had learned computer skills in junior high were devastated when they got to senior high school and found too few machines to go around. Says Alpena Elementary School principal Burt Wright: "I've got high school kids begging to come in after school and use our machine." The truly addicted-known half scornfully, half admiringly as computer nerds-may drop out almost entirely from the everyday world. In Lexington, Mass., one legendary 16-year-old nerd got so deeply immersed in computers that he talked to no one, headed straight to his terminal after school and barely sat down for meals.

The only way his father could get him away from the terminal was to go down to the cellar and throw the house's main power switch, cutting off all electricity. Barry Porter, 14, of San Francisco, is a computer-age truant, so attached to the machine that he often skips school, rarely reads anything other than computer manuals and hangs out with his pals in the Market Street computer store, often plotting some new electronic scam. Barry (not his real name) currently boasts an illicit library of about 1000 pirated (i.e. illegally copied) programmes worth about $50,000 at retail prices, including such software gems as VisiCalc, the popular business management and planning programme. Before security was tightened up, he regularly plugged his computer into such distant databanks as the Source (which provides news bulletins, stock prices, etc.) via telephone without paying a cent.

BUYING PLACE

The personal computer industry has grown as a direct result of the evolution of the microprocessor. This evolution began when Intel in 1971 packaged a complete, if somewhat limited, processor with a 4-bit word size in a single integrated circuit. The company followed with the first 8-bit processor in 1972 and an improved version in 1974. One year later Micro Instrumentation and Telemetry Systems, Inc. (MITS), an Albuquerque firm, developed the first personal computer around Intel's 8-bit processor. The basic system sold for $395 in kit form and $621 in assembled form, not including accessories (peripherals).

Though the MITS system is no longer manufactured, its method for connecting peripherals and the main computer has become an industry standard. Within three years after it was introduced, Radio Shack, Apple, and Commodore had entered the market. Now makers of large computers such as IBM and Honeywell, as well as minicomputer leaders such as Digital Equipment Corp. and Data General Corp., are also making personal computers, having observed their traditional markets being eroded by the new low-priced products. The results are rapid development of the market and growing popular interest and faith in personal computers.

The pioneering manufacturers did not survive beyond the initial phase of personal computer development resulting from the pre-occupation with the needs of hobbyists. New entrants such as Radio Shack, Commodore, and Apple captured a major share of the market in 1978 by promoting fully assembled, ready-to-operate systems that were easier to use. Though the success of the companies now in the competition will depend at least partly on their financial and technical resources, their products' ease of use (or "user-friendliness," in trade jargon) is a significant competitive feature.

The personal computer market can be divided into four segments in terms of the computer's intended use: business, home, research-technical, and educational. The business segment, by far the largest at present, accounted for 750,000 sales (retail value of $2 billion) in 1982, or 54 per cent of the units sold and 65 per cent of the dollar sales. This segment will continue to be a major factor, with more sales than all other segments put together. No wonder, then, that the leading makers of personal computers are concentrating their sales efforts on the business sector.

The most visible segment of the personal computer "revolution" is in homes, where computers are used for entertainment and education, sending messages, and home finances. The development of appropriate software will soon enable sophisticated users to pay bills, manage bank accounts, compute taxes, and even buy household items without written transactions.

The potential market for personal computers in education has barely been tapped. Computer manufacturers recognize that schools are the logical environment in which to develop computer skills, and they have reason to expect that students will purchase the machine on which they first learn computing. So manufacturers are offering many price incentives to the educational sector. Yet the educational sector will continue to account for only about 15 per cent of total sales of personal computers for the rest of this decade, as school funding will be limited.

Although they want to serve all four segments, the industry leaders (Apple, Radio Shack, Commodore, IBM, and Xerox) are focusing on the business market. Vector Graphics is also emphasizing this market by tailoring personal computer systems to the needs of particular industries. Atari (Warner), Intellivision (Mattel), and Texas Instruments are focusing on the home market. Hewlett-Packard is offering a highly specialized compact product with built-in printer, tape memory, and video monitor especially for the scientific community.

In general, both prices and manufacturers' profit margins are failing, and mass marketing is becoming the rule. The major competitors win each spend over $10 million in 1983 on advertising aimed at expanding the market and establishing their place in it. The dynamics of the entire personal computer market may be changed substantially by new entrants. Traditional barriers to entry into this field have crumbled. For example, manufacturing capability is not essential: the IBM personal computer is assembled almost entirely from pre-manufactured components not of IBM origin.

By contrast, strength in marketing and distribution is a significant advantage, and many Organisations with this capability may seek to replicate IBM's strategy for rapid entry into the field. Organisations such as General Electric, Procter and Gamble, Phillip Morris, and Du Pont all have the ability to enter the personal computer market in this way.

SELLING PLACE

Strategies for marketing mainframe computers are not appropriate for marketing personal computers—profit margins are not large enough to justify hiring internal sales forces to sell directly to end users. As a result, producers are now experimenting with a wide variety of distribution strategies.

Franchised retail chains constitute a major distribution channel. For example, the Computer-land chain sold $200 million worth of computers and related accessories in 1981. These stores distribute the products of many vendors, and their volume is large enough to support a technical and maintenance staff. Manufacturer-owned retail stores have been used successfully by Radio Shack. Other computer manufacturers like IBM, Digital, and Xerox have also opened such stores but only to supplement existing distribution channels. Except for Xerox, each manufacturers stores sell only that manufacturer's products, and a prospective buyer is thus obliged to visit several stores to compare equipment of different makes.

Department store outlets have generally been unsuccessful. Mass merchandisers depend for profits on fast-selling commodities. According to computer industry data, personal computer buyers make four shopping trips totalling as much as seven hours when selecting their machines. These buyers also expect sustained support and maintenance services that department stores are unaccustomed to providing.

Office equipment stores specializing in copiers, typewriters, word processors and other office equipment are well positioned to reach the most promising future market for personal computers and auxiliary equipment. If these stores can provide adequate servicing, they will become very popular.

Consumer electronics stores such as Tech Hi-Fi have been marketing personal computers with some success, but the lack of expertise at the store level has been a constraint. Japanese manufacturers have also established ties with such stores as distributors of other Japanese products. These channels will become a major factor as the Japanese increase their share of the personal computer market. Independent retailers often lack the capital required to compete vigorously and are therefore not gaining in numbers and importance as quickly as other retail channels. Catalogue showrooms have been used by Texas Instruments, but personal computers require follow-up support that such showrooms have been unable to provide.

Mail order firms offering discounts ranging up to 30 per cent appeal to price-sensitive customers, but their total lack of continuing support disenchants users. Price-cutting penalizes the full-service dealers on which manufacturers want to rely as major sales outlets. Accordingly, major companies are trying to discourage sales through mail order firms.

Direct sales staffs are used for large-volume sales to government, educational institutions, and major corporations who prefer dealing directly with the manufacturer. But such direct sales tend to antagonize dealers by depriving them of some of their most profitable opportunities, and profit margins are inadequate to support direct sales to small-volume and individual buyers. Value-added houses serve specialized users with coordinated hardware and software, such as printing companies that need word-processing and typesetting capabilities but have little or no internal computing expertise.

Though franchised retail outlets are the largest sellers of personal computers in the US, the one compelling characteristic of personal computer distribution today is diversity. No one form of vendor or market approach is dominant.

Five

FUNCTIONING ELEMENTS

Software—the set of instructions that tell a computer what to do—is becoming more important than the computer itself. The booming software industry is therefore setting the pace of the information technology, revolution and the pace is frantic. This article first appeared in *Business Week*, February 27, 1984. Hardware is no longer where the action is. Computers are becoming remarkably similar—in many cases they are turning into off-the-shelf commodity products. Now the computer wars are being fought on a new battleground: software-the instructions that tell computers how to do everything from processing payrolls to playing video games.

NEW APPROACHES

To complete successfully amid all the turmoil, software suppliers are struggling to formulate new strategies. One common focus is advanced design-especially the development of software aimed at making computers easier to use. But just as important to the success of new software as advanced design is marketing. Nowhere is the new attention to marketing more noticeable than in the hotly competitive personal computer business. To reach the millions of personal computer users, software companies are spending huge amounts to introduce and advertise their products. Industry watchers have dubbed this obsession with splashy promotion "the Lotus syndrome—a reference to the more than $1 million that Lotus Development Corp. spent over a three-month period in 1983 to launch its first product, the highly successful 1-2-3 package. "Lotus advertised so much that companies are going to be forced to step up their advertising just to be heard above the noise," says David E. Gold, a San Jose (Calif.) computer consultant.

Advertising, in fact, has grown so important to software success that it has become one of the biggest barriers keeping out new companies. "The cost of technology development is dwarfed by the marketing cost; the ante has really been upped," says Rodney N. Turner, vice-president for sales at Ashton-Tate, a Culver City (Calif.) software company best known for its dBase II package. It takes as much as $8 million to launch a new software product today, he estimates. Ashton-Tate, by contrast, was founded on shoestring-$7,500. Any personal computer programme must be carefully packaged and promoted. "It has to look good and has to be well supported by national advertising," maintains retailer Gregg E. Olson, a salesman at Mr. Software in Boulder, Colo. "A company that wants to sell to us has to have all their marketing elements in place to establish credibility with us and our customers."

Even companies that write software for the large minicomputers and mainframes are plowing more money into marketing. While programmes for the larger systems, unlike those for personal computers, do not require heavy consumer advertising, says Martin A. Goetz, senior vice-president at Applied Data Research Inc., his company still boosted its 1984 advertising budget by 60 per cent over 1983, to about $2 million.

This emphasis on marketing comes chiefly from the need to reach a different, far broader group of potential customers. "Ten years ago, [our customer] was the data processing department, and if you had a product, you sold it to the technicians," says Robert D. Baskerville, group vice-president for product management at Computer Sciences Corp. in El Segundo, Calif. Today, he notes, a software company has "to sell to end users, and you have to emphasize more than the technical capability—you have to really sell the benefit."

Software marketing now means beginning with detailed product planning, so that a programme will better meet the needs of customers. Before Wyly Corp. even began a $20 million programme to diversify into applications software, for example, it brought in all its salesmen from the field to tell programmers what customers wanted-something most software companies had never done. "With modern software development techniques, you can produce almost perfect code, but if you don't understand the market, it could still be useless," says Ron W. Brittian, Wyly's vice-president for research and development.

Critical, too, is a reputation for good service and customer support. "To grow from a startup into a large software company is not so easy anymore; it's a question of distribution and of educating the customer in how to use the product," says Anthony W. Wang, executive vice-president at Computer Associates International Inc. Toward this end, his company recently installed a $250,000 television studio—costing $200,000 or more a year to run—to make training videotapes for its software customers.

HELP BY CUSTOMER

As competition heightens and customers become more demanding, companies selling personal computer software are providing a level of customer support far greater than anything they offered before. MicroPro International Corp., for example, has plowed $2.5 million into developing computer-aided instruction and retail-support aids for its software, which includes the best-selling WordStar. The company is also hiring journalists and other non-technical writers to produce more readable instructions for today's less sophisticated customers. "A product is not a product until you have computer-aided instructions, it's not a product until you have, video instructions, it's not a product until you have honest-to-God understandable language [in the manual]," says H. Glen Haney, president of MicroPro.

Establishing a marketing presence is especially critical to companies that provide personal computer software, because they must win space on already crowded retail shelves. Computer retailers are reluctant to take on new software without feeling confident it is a winner. "Unless we're absolutely convinced that it's a great programme, we'll wait to see how other channels of distribution do with it," says John H. Rollins, national manager for Sears Business Systems Centres, the computer retailing arm of Sears, Roebuck & Co. One group trying hard to enter the software business may have an edge: the traditional book publishers. Because they already sell books to computer stores, the publishers "provide ready-made distribution channels," says Input's Freeman. Agrees William M. Graves, president of MSA: "There is a tremendous similarity in the microcomputer business and publishing."

WIDE RANGE OF PRODUCTS

No matter how they distribute their products, software companies are finding that they must offer a broader range of products than ever before. "The strategy is to offer a complete solution," asserts Robert N. Goldman, president of Cullinet. "If a customer ends up with eight different vendors, none of the software works together." Cullinet, for instance, is augmenting its data base management software with such applications as general ledger accounting and manufacturing control programmes.

Broadening a software line also helps a supplier leverage the large amounts of money spent on establishing a reputation and brand recognition. VisiCorp, which gained market fame with its hit programme VisiCalc, is trying to expand its applications software with its family of Visi On integrated software. Similarly, Microsoft is plunging into such applications software as word processing and financial analysis to build on its reputation as a supplier of personal computer operating systems.

The trend towards comprehensive offerings will force small companies to target well-defined market niches. There will continue to be demand for specialized packages designed for the needs of a particular industry or profession. "There will be several very large software companies," contends John P. Imlay Jr., chairman of MSA, "but there will be literally hundreds of small companies, with under $100 million [in sales], that have specialized market niches."

Perhaps the most fundamental change in the software industry is the blurring distinctions among the suppliers. No longer can they be neatly divided into companies that make basic systems software and those that write programmes for specific applications.

Moreover, the top mainframe software companies are rushing to market with software for personal computers. And vendors serving only the personal computer market are joining with suppliers of mainframe software. VisiCorp, for example, recently teamed up with Informatics General Corp. to offer Visi-Answer, a programme that allows a user of an IBM Personal Computer to retrieve information from an IBM mainframe data base. These changes resulted in large part from the growing number of customers linking personal computers to large mainframe systems.

AREA OF DATA PROCESSING

As the emphasis in the data processing industry shifts to software and, as software companies strengthen their sales, service, and distribution, the big-system makers, too, are scrambling to do more to provide their customers with software. "In the old days, our customer wrote his own application [software]," notes Jon Tempas, vice-president for software products at Sperry Corp.'s Computer Systems operation.

Today, "there's an increased expectation for hardware suppliers to provide the complete solution." That means the equipment makers will need to provide more of their own software. Sperry, for example, now writes 95 per cent of the software it sells for its computer line.

Most of the big-system companies, however, are turning to software specialists for help, since much of the demand is for applications software finely tuned to specific industries. "I don't think there's any hardware manufacturer that can-or should-provide all the software," says John E. Steuri, general manager of Information Services Business Unit, IBM's new independent software group. "We'll be more and more dependent over time on software developed outside the company." The pressure to team up with independent software companies is especially strong among makers of personal computers.

"Without an adequate software base, a microcomputer dies," says Eugene W. Helms, vice-president for business development at Texas Instruments Inc.'s Data Systems Group. So TI is recruiting software suppliers to adapt their best-selling programmes for its Professional Computer. Perhaps the most extensive effort to sign up independent software companies was made by Apple Computer Inc., which courted more than 100 companies to write software for its new Macintosh computer. The California company was successful in signing up more than 80 of them.

GROWTH OF THE WIDE RANGE

Companies that do not move quickly to develop a broad line of software will have trouble keeping up. Consider

Tymshare Inc., a computer timesharing company. As computer prices began to drop precipitously, more and more of Tymshare's customers stopped renting computer time and purchased their own machines. Tymshare did not have any software packages to sell to its former clients. "We used to look at software as simply an in-house tool that we needed to offer time-sharing," says one Tymshare executive. That attitude, he admits, "came back and bit us in the rear end." In 1983 the California company lost $1.7 million while sales fell 3 per cent to $288 million. Now it is engaged in a major effort to expand into applications software, through internal development and by licensing packages developed by other companies.

With most software companies trying hard to move in the same direction, everyone will face stiffer competition. Already, prices for personal computer software are sliding wherever similar programmes have proliferated-in electronic spreadsheets and word processors, for example. And a big battle is under way among suppliers of the various "windowing" software packages—software "environments" that permit the users of personal computers to display several tasks at once. VisiCorp has been forced to slash the price of its Visi On environment package from $495 to $95. "The consumer is going to force the prices down simply by demand," asserts Alvin B. Reuben, executive vice-president at Simon & Schuster Inc.'s electronic publishing division.

As prices fall, the opportunity for, newcomers to jump in and grab quick and easy profits will all but disappear. The cost of developing and marketing new programmes is mushrooming just when margins are shrinking. VisiCorp successfully launched VisiCalc in 1978 with a $500 budget. But the California company has spent more than $10 million developing its latest product, the Visi On environment. Product life is also getting shorter as new products come out faster. As a result, says Softsel's Wagman, "the stakes have gotten higher and much riskier from a development point of view."

For many companies, the risks will ultimately prove fatal. "There are companies out there that are already feeling the pinch of increased costs, new product announcements, and a changing marketplace," warns consultant Gold. Industry watchers predict that some big-name failures will occur within the next 18 months. By the end of the decade, many experts expect the software industry to have consolidated its thousands of suppliers into a few major players.

For those that make it, the future is promising. The demand by the growing army of computer users for easier-to-use software to handle an exploding variety of tasks will drive the industry to create software with far more capabilities than anything available today. "They only limit," says Stuart A. Walker, vice-president of marketing at Konware Inc., "is the limit of new ideas."

Software in the Office: One of the main reasons software has become so highly visible is that the computer has moved from the back room into the office.

Programmes of the past, buried inside corporate data processing centres, primarily handled clerical and accounting tasks, such as turning out payrolls and keeping track of accounts receivable. Today's computer handles a rapidly increasing variety of management tasks—employing information as a competitive marketing weapon, for example.

That trend, coupled with the explosion in the use of personal computers by executives, has given new prominence and importance to software in us corporations. "The Programmes that we install to do our company's business are analogous to the tools an auto maker puts in place to manufacture auto parts," says Jeffry A Alperin, an assistant vice-president in the Information Systems Support Dept at Aetna Life & Casualty Co. "Our plant is our data Processing system." Since the same computer hardware is available to everyone it is the software that often gives a company a competitive edge. At oil companies, software is now helping drive the search for oil and gas," says Michael C. Balay, general manager of information technology at Gulf Oil Corp. "You can have the same [seismic] information, but a unique software package may help you better interpret the data. "

Nowhere is this increasing importance more obvious than in corporate data processing budgets. Belay estimates that Gulf spends about half of its annual data processing budget to write its own software and to buy packages from independent suppliers. At Aetna, half of the 4000 people on the data processing staff are programmers. And Alperin says that 4 per cent of the $237 million that Aetna spent on information processing last year went to purchase packaged software.

Moreover, individual owners of personal computers are no different from large corporations: over the life of a computer, its owner will spend $2 on software for every $1 spent on hardware, according to industry estimates. But users are finding it increasingly difficult to produce their own programmes. Writing software is still a time-consuming process, and users find that, while they want to employ their computers to handle more applications they do not have enough programmers, time, or money to write all of the necessary software in-house. In fact, at many companies users must now wait as long as 18 months for new programmes to be written. "The end users are demanding faster development of solutions," declares Richard R. Douglas, group vice-president for the US Marketing and Services Group of Honeywell Inc.

READY-TO-RUN SOFTWARE PACKAGES

To come up with this software faster, computer users are compromising-buying more ready-to-run software packages than ever before. "There has been a turnaround from five or six years ago," says James T. Manion, sales vice-president at ASK Computer Systems Inc. Few computer users write all their own programmes anymore, he says. But while users sacrifice some uniqueness by buying a software package, that disadvantage is often outweighed by the time and money saved. A package can cut the time it takes to get a system up and running by more than half, figures Mayford L. Roark, executive director of systems at Ford Motor Co. He also estimates that buying a programme from an outside vendor can cut development costs by 30 to 75 per cent. "It's almost impossible to think of a function you might want to do on a computer that you can't find several software packages for," says Roark.

Where a custom solution to a problem is required, companies often find they can save time by modifying a standard software package for their own use. That is what Aetna did when it needed software to offer a new type of life insurance called universal life, which features flexible premiums. That allowed the company to match its competitors' universal life offerings quickly. With such a variety of software available, companies are finding that writing their own programmes from scratch makes about as much sense as drawing their own road maps for their salesmen.

US corporations brought more software packages last year than in the previous year, says International Data Corp., a market researcher. Such purchases, along with software services, now account for more than 8 per cent of the average data processing budget, up from 6 per cent in 1980, IDC says. Similarly, purchases of personal computer software packages soared 74 per cent last year to top $1 billion, according to market researchers at California's Input.

BEHAVIOUR OF CUSTOMER

The customer's new attitude towards software is beginning to alter fundamentally the way that computer systems are sold. "We're selling to a much more sophisticated consumer base than in the past," points out Elizabeth M. R. Hall, product manager at Information Science Inc., a Montvale (NJ) software company. From now on, adds Honeywell's Douglas, "Software looms as a much more important factor in [a user's purchase] decision than hardware."

Computing used to be the solitary domain of a priesthood of programmers. The arcane languages that they employed to command their giant mainframe systems were shrouded in such 'complexity that few laymen could understand, much less control, these behemoths. But the inexpensive yet powerful personal computer is changing all that. Millions of non-technical users are now running computers. These new operators, however, refuse even to read an instruction manual, let alone memorize the cryptic commands of the priesthood. Their aversion, coupled with the dramatic slide in the cost of computer power, has created a revolution in the way software is written.

Today, making a programme simple to control and easy to use is just as important as what the programme actually does. Says Jeanne M. Baccash, a software engineer at American Telephone & Telegraph Co.'s Bell Laboratories: "There's a whole new thrust to reach a market that doesn't know or care what it means to 'boot a system' [start a computer].

These efforts to make software simpler are crucial if the information processing industry is to continue its fast growth. "Rapidly advancing technology has left consumers, trainers, and computer salesmen behind," says Terry L. Opdendyk, president of VisiCorp. "How to spread the word of how to use the products is the key limiting factor [to the industry's growth]."

Man-machine Communications: To reduce the number of commands that a user must memorize and type into a computer to get it to work, the latest software enables users to communicate with the machine in new ways. Some of these techniques are relatively simple, such as establishing menus to show users what commands are available to choose from. The most elaborate methods, used on such machines as Apple

Computer Inc.'s new Macintosh, replace commands with an array of tiny icons or pictures—for example, a file folder to indicate filing. To tell the computer what to do, users point to the appropriate picture on the screen by moving a pointing device called a "mouse" over a desktop.

Data Exchange: Early programmes were built to accomplish tasks, such as calculating financial forecasts, with only the data located within the user's own computer. But users have grown more sophisticated and now want to get data stored in other machines—say, the corporation's central mainframe system.

Artificial Intelligence: This embryonic method of programming, which enables software to mimic human thought more closely, might one day be the easiest approach to use. It is starting to show up in programmes such as Artificial Intelligence Corp.'s Intellect and Microrim Corp.'s CLIO, which let a user ask a computer for data with English sentences rather than esoteric commands.

Most programmers are ill-equipped to figure out how office workers and other non-technical users best handle computers. So companies are bringing in professionals from disciplines as far afield as education and psychology to help in the design of man-to-machine communications. AT&T, for example, had a staff of psychologists survey about 400 computer users to help it decide how to add commands to its Unix operating system. Designing easier-to-use software also requires new development techniques. For instance, International Business Machines Corp. now tests its new software in "usability labs." In these labs, volunteers try to use IBM software while researchers with video cameras watch from behind one-way mirrors. "In the long haul, those that have easy-to-understand software will be the successful companies," says John E. Steuri, general manager of IBM's Information Services Business Unit.

A lot of discussion is going on in research circles over how best to use menus, pictograms, and other techniques to help both computer novices and experts. The type of coaching that a neophyte computer user requires becomes annoying once the user has learned to use the software. So designers want "to make sure the user is not presented with extraneous information when he has to decide what to do next," says Brian K. Reid, a Stanford University electrical engineering professor. Micropro International Corp., for one, is trying to solve this problem by using a time-delay activator.

If a user types in a command less than two seconds after being asked for it, the programme assumes the user is proficient and skips over the menu listing of options. Users who take longer are given the benefit of directions from a menu. Researchers are also struggling with the problem of how to standardize commands. Each computer programme now uses different commands to accomplish the same function. An early attempt to solve this problem is the "environment" or "windowing" software packages for personal computers that are now coming to market.

These packages-products such as Microsoft Corp's Windows and VisiCorp's Visi On-divide the computer screen into segments, each of which shows a different task. Common functions are performed the same way in each window. The large amount of this kind of innovative software now being developed to make personal computers easier to use is also forcing the mainframe software companies to follow suit. "Mainframe vendors have been forced to make software easier' because users have been spoiled by micros," says David Ferris, an industry consultant.

DATABASE ADMINISTRATION

Information Builders Inc. has developed a version of its Focus data base management software that can run on an IBM Personal Computer. But the PC version, called Pc-Focus, "has more features than the mainframe product because after using a personal computer customers have come to expect more," says Gerald D. Cohen, president of the New York company. One of the Pc-Focus features is called Table Talk. Each time a user moves from one step to another in the course of retrieving information from a data base, the computer automatically presents him with a menu of the appropriate choices.

Another software challenge is to find simpler ways to exchange data between programmes. In most cases, swapping data between a large mainframe and a personal computer is difficult because the two units use different formats for storing their data and different commands for retrieving them.

The problem is similar to that of a person who is trying to communicate with 10 people, each of whom speaks a different language, explains Robert J. Spinrad, director of systems technology at Xerox Corp. "I could either learn all nine other languages," he says, "or we could all learn a common one-say, Latin." But software vendors are not waiting until everyone in the industry agrees on what common language all computers should learn. In the past year a host of vendors, including Cullinet Software, Informatics General, and Cincom Systems, have announced products that link personal computers to large mainframes. Such capabilities will be a requisite for any software in the marketplace, says President Frank H. Dodge of McCormack & Dodge Corp. "If mainframe software doesn't allow that link," he adds, "it's going to be pushed aside pretty rapidly."

COMPLICATED FILE MANAGEMENT

Some experts think the ultimate mechanism for exchanging data is the use of sophisticated file-management software, often called data base management systems. These large programmes index information and then store it in such a way that it can be retrieved using a variety of names-much the way a library card catalogue lists the same book by author, subject, and title. By providing this uniform filing structure, a data base system simplifies the exchange of data.

What may soon make computers even easier to use is artificial intelligence (AI) software. Some companies are already working on AI programmes that will eventually be able to remember an individual's habits in using the computer. The first AI applications from Microsoft-expected within the next year-will be likely to use these rudimentary pattern-recognition techniques in tutorial programmes that teach novices how to operate software. These programmes will adjust the level of tutorial difficulty by determining the proficiency of the student running the programme. Says Microsoft Chairman William H. Gates: "Just as humans take actions based on past experience without having to be told again, and again, so will software.

As the second-largest group of computer companies in the world-after the US—they exported $2.7 billion worth of computers in 1983. But in their single-minded determination to build and export hardware that is faster and less expensive, the Japanese have given short shrift to software development. "Hardware manufacturers have been lazy about developing software," acknowledges Hisao Ishihara, managing director of the Japan Software Industry Assn. But now, he points out, "that is suddenly changing." Faced with the twin problems of rapidly falling hardware prices and the growing percentage of computer budgets being spent on software, Japanese hardware makers are scrambling to shift their resources into software development.

RESEARCH AND DEVELOPMENT

In 1981, for example, Hitachi Ltd's Computer Division spent just 10 per cent of its research and development budget on software. In 1984, even though software accounted for about 10 per cent of its computer revenues, Hitachi spent 30 per cent of its R&D money on software. "These days, more and more of the value added in information processing comes from the software," says Toshimitsu Kaihatsu, head of Toshiba Corp.'s Software Management Dept.

The Japanese are also keenly aware that to continue to be competitive internationally in computers they must now start developing world-class software. "We have to sell more overseas if we want to recover our software development costs," comments Shoichi Ninomiya, general manager of Fujitsu Ltd.'s Information Processing Group. His company spent close to $100 million a year-more than one-third Of its R&D budget-to develop software for its mainframe computers. But almost all of that software is now sold to Japanese customers. Until now, software has been the biggest handicap the Japanese have had in selling their equipment abroad-especially in the office and personal computer markets. This point was driven home in 1982 when International Business Machines Corp. sued Hitachi, one of the largest Japanese exporters of computers, charging the company with copying IBM software and reselling it with Hitachi's machines.

Hitachi agreed in an out-of-court settlement to pay IBM between $2 million and $4 million a month in software license fees and agreed to let the US computer giant inspect all new Hitachi products before they go on the market to ensure that they do not infringe any IBM copyrights. To avoid a similar lawsuit, Fujitsu has also agreed to pay IBM many millions of dollars and promised not to copy the US company's software, although it would not disclose the details of its agreement. Because of language and cultural differences, software written in Japan is obviously difficult to export. Not only must instruction manuals be translated, but often the programmes have to be completely rewritten. Japanese accounting rules, for example, are different from those in the US, so accounting software written in Japan is useless in the US. This shortage of programmes has severely limited the export potential of Japanese home and personal computers.

Japan's software industry is also held back by the preference of Japanese buyers for custom software rather than the standard software products that US customers are increasingly buying.

SOFTWARE FOR INDUSTRY

Toshiba recently completed a factory that employs 3000 software engineers to develop industrial software. Now the company is building a second software factory that will employ 2000 more programmers. "To overcome Japan's language problem and compete with the US," says Toshiba's Kaihatsu, "we have to have productivity double that of the US." NEC Corp., which already spends a significant portion of its annual $400 million software budget on productivity tools, says it will use its productivity and quality advantages to crack the US market. It has hired US software engineers to analyze the needs of US computer users.

The resulting lists of requirements are then fed into computers at NEC's Japanese software factories, where programmers write the software. In this fashion, NEC has already started developing commonly used business applications, says Yukio Mizuno, an NEC vice-president. Perhaps the best illustration of the importance the Japanese now attach to software is the move by the Ministry of International Trade & Industry (MITI) to back the industry. MITI has set up several research laboratories to work on software, including a lab that is developing the so-called fifth-generation computer and software. That lab is budgeted to receive $23 million in MITI support this year. The ministry is also giving low-interest loans and tax breaks to software developers.

MITI hopes to boost its fledgling software industry further with a proposed revision in the copyright law.

The ministry is pushing to allow Japanese companies to save money and programming time by making it legal for them to copy portions of existing software products without the permission of the original developers.

Industry observers say the law, if passed, would help the Japanese leapfrog US software companies by enabling them to copy popular US programmes and incorporate them into Japanese software products. But even with MITI's help, it will be a long time before Japan's software industry catches up with its US competitors-if it ever does. In 1982, software sales in Japan were only $1.4 billion-just one-quarter of the total US software sales that year.

SOFTWARE INDUSTRY IN JAPAN

As a result, an independent Japanese software industry has been slow to develop. And the computer manufacturers' efforts to develop all the software themselves have been crimped by the large sums they must invest to improve methods of getting information into the computer and processing it, using Japanese *kanji* characters. But it would be dangerous to write off the Japanese as competitors in world-class software. "The Japanese are as dedicated to computers as they were to autos and shipbuilding," says John P. Imlay Jr, chairman of Management Science America Inc. "If US companies do not innovate and supply quality products and customer support, the Japanese could make inroads."

Already, Japan has a big chunk of the US market for the simplest type of software, that written for video games. Cultural differences are not a barrier to the export of such hit games as Pac-Man. The fast-growing software markets for engineering and scientific applications-which manipulate schematic drawings and numbers rather than words-are other areas where Japanese software could be sold overseas with little modification. The Japanese have also demonstrated their technical ability by creating powerful supercomputer software and sophisticated banking and airline reservation systems. "People are misreading the capability of the Japanese when they say Japanese can't build good software," maintains Joseph C. Berston, president of Comstute Inc., a software consulting firm based in Japan.

Indeed, Japanese companies may actually have some advantages over their US rivals. Because of the legendary thoroughness of Japanese workers, “the finished product here is better, more reliable, and easier to maintain,” says consultant Berston. Labour costs are also lower for Japanese software makers. Well-educated, highly disciplined Japanese programmers are paid an average of about $10,000 a year. Their US counterparts can expect a starting salary twice that—from $19,000 to $25,000 according to a survey by Robert Half International Inc., a personnel agency.

In addition to that salary differential, the Japanese claim their programmers are 10 to 15 per cent more productive than their US counterparts because of Japanese investments in programme development aids. To widen that margin, they are now building software factories that give their programmers access to even more sophisticated tools.

GROWING DEMAND

A key reason for the change in emphasis is an overwhelming demand from customers for packaged software that will let them apply computer power to a broad range of new tasks. Increasingly, corporations are finding they do not have the resources to write the programmes they need. As a result, most companies have stopped writing their own software and are instead buying standard software packages. So far, producers have been unable to keep up with the need, and there has been a severe shortage of software able to take advantage of the power of the latest machines. “We’re getting the hardware we need from the industry, but software has not moved out at the same rate,” says Robert J. Metzler, vice-president of First Computer Services, the data processing arm of First Union National Bank in Charlotte, NC. “There’s a tremendous cry in the industry [for software].” The flood of personal computers pouring into small businesses and homes has created an even faster-growing new market.

The booming demand for new and better programmes has quickly turned software into big business. During the 1970s software was stiff a cottage industry, 'with sales totalling just $2.7 billion annually. This year-according to estimates by Input, a California market research firm-sales are expected to top $10 billion.

With the industry growing that fast, several once-small software companies have become sizable corporations. Management Science America Inc. (MSA), for example, the largest independent software supplier, quadrupled in size during the past three years, topping $145 million in sales in 1983. Trying to capture some of the soaring market, thousands of new companies have entered the software business in recent years; by one count there are more than 3000 software companies now.

While over the long term a shakeout is probably inevitable, the near-term outlook for most of these companies appears extremely bright. "The upside potential has barely been tapped," says Robert M. Freeman, senior analyst at Input. He expects the market to keep growing by a dizzying 32 per cent a year, topping $30 billion in 1988. Software sales of that magnitude would amount to half of the $60 billion hardware business expected for the same year; today, revenue from software is equal to only 27 per cent of the value of all the computer hardware sold.

SALES OF SOFTWARE

Sales of software for personal computers-the fastest-growing part of the software industry-should rise by an astounding 44 per cent annually over the same five years. "We're finally at the point where software applications are going to be a big moneymaker," says Jack M. Scanlon, vice-president of the Computer Systems Division of AT&T Technologies Inc. But with the promise of such rapid market growth, has come a feverish competition that is beginning to restructure the entire industry.

Until recently, nearly all software companies concentrated on a particular niche and fit neatly into one of three distinct market segments that were divided along the same lines for both main-frame and personal computers.

ACTIVITY OF HOUSEKEEPING

Systems software, which handles basic housekeeping operations such as controlling the printer and memory, was supplied primarily by the computer makers. They were joined by independent software companies in providing utility software, which, among other things, helps programmers write programmes. And a host of other independent software companies competed in the third market; applications packages, which tell a computer how to carry out specific tasks such as accounting payroll or word processing.

But those distinctions are blurring:

Consolidation: International Business Machines Corp., to move faster into applications software, placed all its software efforts into a single, entrepreneurial unit—the same type of Organisation it used to launch the highly successful Personal Computer.

Acquisitions: To provide a full complement of software for their computers, makers such as Hewlett-Packard, Burroughs, and Prime Computer are rushing to buy applications software companies. Other manufacturers, including Honey-well, Sperry, and Digital Equipment, are setting up joint ventures with software suppliers. Says Roger T. Hobbs, vice-president for software products and services at Burroughs Corp.: "The demands for software are increasing so rapidly that it is impossible for the manufacturer to keep up."

Expanding Product Lines: Software companies are expanding product lines to maintain their competitive edge. Those that supply programmes mostly for large mainframes—MSA, Cullinet Software, and Computer Associates International, for example—are snapping up personal computer software houses. And companies that specialize in systems software for personal computers-among them, AshtonTate, Digital Research, and Microsoft—are adding general-purpose applications software to their product lines. "Today you have to have a broad line," says Terry L. Opdendyk, president of VisiCorp. Otherwise, "the vendor is increasingly vulnerable to competition."

New Players: To grab a piece of the action, publishers and other communications companies outside the computer business—including CBS, Dow Jones, Dun & Bradstreet, McGraw-Hill, and Simon and Schuster-are licensing programmes they then sell through their own distribution channels. "It's a very fragmented industry, but the potential is absolutely huge," says Richard W. Young, president of Houghton Mifflin Co.

Japan's Drive: To catch up with their US rivals, Japanese computer makers are launching major software development efforts. Perhaps the best evidence of just how important software has become is the drive by Japan's Ministry of International Trade & Industry to change the law to boost Japan's fledgling software industry. One result of many of these trends is that mergers and acquisitions in the industry are at an all-time high. In 1983 there were 146 acquisitions, valued at more than $1 billion-up 130 per cent from 1982—reports Broadview Associates, a New Jersey company handling such transactions. And this frantic pace of activity is turning the software business into a hot investment area.

Six

ROLE OF COMPUTERS

Recent years have seen both a decline in the price and a rise in the power of personal computers. You can get a fairly basic word-processing kit, including a monochrome monitor, dot matrix printer and word-processing software, for around £400; Whereas £1,500 would buy you a powerful and sophisticated IBM-compatible personal computer (PC) with a built-in integrated suite of software (word processing, database, spreadsheet, graphics, communications, etc.), DOS and Windows operating systems, colour monitor and a bubblejet printer.

There may be a temptation for a community information service to assume that a computer is now a necessity for its work, before asking the obvious question: 'What do we need one for?' Consideration also needs to be given to the question of who is expected to use the computer. If your service relies on volunteers, they may have an aversion to new technology or simply not work long enough to become familiar with and confident in using a computer. Before plunging into the murky waters of information technology (IT), you should first of all ask the kind of basic questions that needed to be addressed when setting up your service, such as 'Why do we need IT', 'What functions do we want it to perform?', 'What system is going to meet our needs?' 'Who is going to use the system?', 'Where can I get advice?', 'What training will be required?', and so on. There are many reasons why an information service might want a computer but they may not necessarily all be the right reasons. Some wrong reasons are:

— Everybody else has got one;
— To improve the image of the service;
— To sort out our database /statistics/ financial accounts, which are in a mess.

This latter brings to mind the computer cliche 'garbage in, garbage out', namely that if you put rubbish into a computer, you can expect to get rubbish out. Before inputting any information into a computer, you must first have a logical and organized manual system. The computer won't do it for you.

Some right reasons, might be:

— To provide faster and more timely access to information;
— To facilitate access to a wider range of information;
— To provide more sophisticated access to information;
— To enable more people to have direct access to information.

Before making a decision on what equipment to purchase, consider what functions you want the computer to perform. These might be one or several of the following:

— database management;
— recording statistics and, possibly, producing these in graphic form, i.e. as bar or pie charts, graphs, etc.;
— running welfare benefits and other programmes;
— running a local view data system;
— online access to other external databases;

— word processing;
— financial accounting;
— desktop publishing.

If you require most of these functions, a fairly powerful machine would be necessary but you might be better advised to consider more than one computer or even linking them in a local area network (LAN). A LAN enables personal computers to communicate with each other and share resources such as software and printers. Usually, a LAN is driven by a computer with a large hard disk which contains the software programmes and the data input by the rest of the PCs or terminals on the system. It also provides access to shared printers on the network and to central back-up facilities.

VARIOUS SYSTEMS

Using a PC is not the only way of setting up a community information database. There are other options which may be open to services which are part of a much larger organisation, like public libraries, and are designed to cope with extremely large databases and/or allow wider access. In the early 1970s British Telecom (or The Post Office as it then was) developed a viewdata system called Prestel which enabled users to access thousands of pages of information on all sorts of topics via the medium of the telephone and specially adapted television sets (and later also PCs). The system has never quite lived up to its early promises but is still going strong. A number of public libraries were involved in the early trial of Prestel and recognized the potential of the system, if not of Prestel itself. Eventually, local authorities began to set up their own viewdata systems of information about council services and other local information. The public could access the system either directly from home or through terminals located in libraries and other public buildings. In some instances, the library service is responsible for maintaining the database, in others another department of the local authority. Viewdata systems offer the following:

— Access to large amounts of continually updated information;

— Use of graphics;

— Public online access from home, workplace or publicly located terminals;

— Centralized updating of information.

Some disadvantages are:

— They need dedicated teams to maintain the database;

— Many clientele of community information services would not be able to afford equipment to access the database or would be afraid of using a computer;

— Limitations on the amount of information that can reasonably be presented by the system.

Some public libraries, like Berkshire, are beginning to recognise the income-generating potential of viewdata, with electronic Yellow pages, teleshopping, rental pages, and closed user group facilities. Apart from running viewdata systems, mini or mainframe computers can also be the hosts for community information databases using such software as IBM's STAIRS (Storage and Information Retrieval System). Some libraries set up their databases on a mainframe computer from the start; others, like my own, have transferred from a PC-based system. Some advantages of using a mainframe computer are:

— Almost limitless power;

— Speed of operation;

— Updating from multi-access points;

— Online access from other council departments, the public, etc.;

— Possible links into computer issue systems;

— Feedback from other departments/ organisations using the system.

Disadvantages are:

— They are not particularly user-friendly - no use of graphics;

— It can be difficult to maintain consistency of input and updating with multi-access;

— They take second place to other council programmes, such as payroll.

— You need to 'log on' each time the system is used.

Reference was made above to possible links between mainframe-based community information files and library computerized issue systems. In fact, a number of these systems now offer add-on features for storing community information themselves. The advantages of these are:

— Access to database through every library in the system;

— Public access through OPACs (Online Public Access Terminals) or, in some cases, dial-in facilities;

— Multi-access to inputting and updating database.

Disadvantages are:

— They can slow down issue system;

— They are not usually accessible by other council departments;

— With multi-access, it can be difficult to maintain consistency of inputting and updating;

— As with viewdata systems, information is usually limited.

Inevitably, with a technology that is developing so fast, there will be new systems for presenting information coming along. Glasgow has been experimenting with HyperCard, which allows the user to search through text, pictures, numbers and instructions to gain information about the city. Before long someone is bound to be using CD-ROM or interactive video.

NET SURFING

Another use for your PC might be to link into other remote databases such, as a local authority's viewdata system. To do this you will need' a modem, to link the PC to the telephone network, and communications software, which could be a separate package or part of a suite of software such as LotusWorks or Windows 3.

Most online databases, except for local viewdata, are not appropriate for community information services, but there is one that is: Volnet UK, operated jointly by The Volunteer Centre UK and the Community Development Foundation. Volnet UK is an online information service containing thousands of references to press items, journal articles, reports, books and current research projects on community development, voluntary action, youth affairs, child care and social policy issues, stored on an easy-to-use computer database. For a modest annual fee (00 local community/ voluntary agency, £90 national/ regional voluntary agency, £150 central/local government department) you get unlimited access to the database (no charge per second or minute), a telephone help-line, plus a photocopy service for articles, for which a small charge is made. Details from Volnet UK, Community Development Foundation, 60 Highbury Grove, London N5 2AG.

DIFFERENT PROGRAMMES

Providing welfare benefits information and advice can be notoriously difficult because of the complex legislation, which is why some advice centres and public library community information services are harnessing the power of PCs to help them. These are usually designed for use by advisers although, over the years, some attempt has been made to develop client-operated systems, largely without success.

Welfare benefits programmes contain the rules of the benefits system and present on the computer screen a series of questions. The adviser, in consultation with the client, types in the details and the computer works out whether the client is entitled to benefit and, if so, how much. The results can usually be printed out and given to the client to take away. The main benefits covered include Income Support, Housing Benefit, Family Credit and Community Charge Benefit. The advantages of these programmes are:

— Consistency;
— They do not let you forget little-used regulations or passport benefits, where claiming one benefit, automatically entitles the client to others;
— A novice worker can tackle more complex cases;
— 'What-if' calculations can be done speedily;
— Printouts of results can be obtained;
— Some clients like the computerized approach and are reassured by it.

Some disadvantages are:

— They are only as accurate as the information input;
— The benefits, system is so complex, programmes are bound to have quirks;
— An experienced adviser can usually work faster than the programme;
— They ran lead to overconfidence;
— Some clients can be intimidated by a computer and find it intrusive;
— They are no substitute for experience.

For more details, on the main welfare benefits programmes available and their features, addresses etc., see the *Computanews* factsheet Welfare benefits programmes (see Wow for details of all the fact-sheets in this series).

There are other PC-based programmes and databases available on such areas as debt advice calculations and sources of funding from charitable trusts. Resource Information Services, in conjunction with LASA's Computer Development Unit, has developed a database of advice agencies in London which became functional in autumn 1992. A good way of keeping up with the latest developments in programmes for advice work is to take out a subscription to the excellent bimonthly periodical *Computanews*.

CHOICE OF FIGURES

Collating and manipulating statistics is one area where computers have a distinct advantage. They can do in seconds or minutes what would take hours of work manually. However, if all you want the computer to do is add up and produce totals in broad categories, then you might as well stick to pencil and calculator. But, for detailed analysis of enquiries and graphic representation of information by means of pie or bar charts, graphs, etc., you will need a computer programme. There are two options: either devise your own system using a spreadsheet package or a relational database system linked with a spreadsheet package, or buy a specially designed programme. A lot of the pioneer work in this field has been carried out by LASA's Computer Development Unit which, some years ago, developed CRESS (Client Records and Enquiry Statistics Service) and is currently engaged on developing a model recording system called STATS. Finally, there are many more uses to which a community information service might wish to put its computer, e.g. word processing, accounts, graphics, desktop publishing, but these aspects apply to many other organisations.

MANY OPTIONS

There are several ways in which you might set about choosing a computer, depending on the level of knowledge of staff in your organisation. Let us assume elementary or no knowledge of computers, in which case the options might be the following:

Dealers-may be OK if you know one who can be trusted or is recommended, but computer salesmen, like car salesmen, frequently try to get you to buy a machine of a higher specification than you need and blind you with technical jargon. Also limits your ability to shop around.

Consultants-usually expensive unless you can beg or borrow their services. Have a tendency to recommend separate software packages for each function or even writing a programme to match needs exactly, rather than considering integrated packages which may require some compromise. OK if all the separate parts can work or are not required to work together.

Other organisations in the same sector who use computers in their work may be able to recommend equipment and software from their own experience. However, they may be reluctant to admit shortcomings and the equipment/ software they use may be out of date, given the speed of developments in this area.

Networks who support advice services or community and voluntary organisations, such as Community Computing Network (65 Litchfield Road, Cambridge. CBI 3SP) or the Computer Development Unit of LASA (2nd Floor, Universal House, Wentworth Street, London El 7SA Tel. 071-377 2798). Apart from giving direct advice themselves, they may be able to put you in touch with others who can help.

Within the space of this book, it is not possible to consider all the potential uses to which a community information service may wish to put a computer. I have therefore concentrated on those aspects which, though not peculiar to information services, are a major feature of their work, namely database management (including view data and access to external databases), welfare benefits programmes and enquiry statistics.

ADMINISTERING DATABASES

There are a number of reasons why a community information service would want to keep its information file on computer:

— Beyond a certain point, a manual information file becomes unwieldy to use and update, whereas a computer database offers speed of access and ease of updating;

— Complex searches involving, for example, subject, place and clientele, are extremely difficult and well nigh impossible to carry out with a manual system. A good computer database management system allows searching of the database by a combination of terms;

— Individual records or subsets of the database can be printed out on demand in a variety of formats, e.g. as mailing labels, lists, standard letters;

— Through networking, online or exchange of disks a computer database can be shared with other workers or organisations;

— Customers can be given direct access to the database through public access terminals in your centre(s) or through a viewdata system;

— Entries can be fed into a word-processing or desktop publishing package and printed out as leaflets, directories, handbooks, etc., either directly or, more usually, by preparing a master copy which is then printed by offset litho.

A computer database is made up of two parts—the software, sometimes referred to as DBMS (database management system) and the database file, which is very much like a manual card index system in that it contains individual records made-up of a number of fields (e.g. name, address, telephone number, hours of opening, etc.), into which bits of information (data) are entered (input). Most database systems store data in a highly structured form but there are some exceptions, known as free-form' or text-retrieval databases, which automatically index all significant words in a document.

Some database systems allow information to be viewed or edited either in tabular form, with each record represented by a row and fields as columns, or a record at a time. The tabular approach is fine if the fields in each record are short but is less convenient where you have lengthy fields. This can usually be circumvented by limiting the number of characters in each field, but then you cannot see the record as a whole. There are basically two types of database management systems: a simple DBMS produces a single file (sometimes called flat file) of records that have a predetermined number of fields and field lengths. This does not mean that field lengths cannot be varied or fields added or deleted, but any changes affect all the records in the file. A flat file may be adequate, say, for a database of community organisations, provided you have reasonably anticipated all the fields and field lengths that you are likely to require.

However, it will not be suitable for a situation where you want to create several different files and have the ability to link them. For example, in addition to the database of community organisations, you may want databases of leaflets, booklets and pamphlets; 'hard' information; individuals who can provide help; and enquiries plus the ability to search on all of them simultaneously. In those circumstances, you are more likely to need a relational DBMS, which would enable you to form links between the different files through the use of common denominators, such as subject or classification fields, client reference number, etc.

Examples of flat file systems are Cardbox Plus (used by Help for Health for its database), Q&A, and FileMaker Pro. Most database functions that come as part of a suite of integrated software, such as LotusWorks, will be flat file systems and may not be sophisticated enough to cope with anything but a simple database. Some of the leading relational systems are DataEase, Paradox, and dBase systems (dBase IV, FoxPro, Clipper). I have used for a number of years a system called Superfile which, at the time, had the advantage of not requiring any programming knowledge from the user. However, most of the database systems have become more user-friendly over the years.

The price you can expect to pay for a DBMS can range from £ 250 to £ 750 but some suppliers offer substantial discounts to voluntary organisations and charities.

CHOICES AVAILABLE

Before choosing a database management system, you should first of all draw up a shopping list of requirements and questions that you need to ask, such as the following:

— Is there a maximum size of file that the system can handle? In most cases this will depend on the size of your computer's memory but some systems may operate more slowly as the database grows beyond a certain size.

— Is there any limit on the size of records, the number of fields in a record, or the size of a field? If so, is this likely to be a problem?

— Is it possible to add, expand or delete fields from a record without difficulty?

— Is it possible to print out all or a selection of records and fields within records in a structured format?

— Can the data be fed into other software packages to produce lists, standard letters, mailing labels, etc.?

— What features does the system provide for searching records? For example, wildcards (* ? =) are a means of truncating search terms to take account of varying forms or spelling, e.g. ORGAN will find any first word in a field beginning with the letters O R G A N, such as 'organize', 'organisation' etc.; 'sounds like' will find records with similar sounding words to the search term, e.g. FAIR and FARE; or Boolean logic will logically link or exclude terms to narrow the field of search, e.g. A and B, A or B, A-G but not C, etc.

— Is there a facility to check on the validity of entered data?

— Is there provision for constructing user-friendly menus?

— How much knowledge is required to set up a database? How long does it take to acquire that knowledge? Can it be self-taught?

— How good is the documentation that accompanies the system?

— Is there a multi-user version of the software?

— What support is offered by the supplier? Is there a user group?

GENERATION OF DATABASES

Having bought your software and installed it on the computer, the next step is to create the database(s). Each database management system will have its own methods for setting up the database and this section can give only a few tips from experience that should apply to whatever system you have chosen. First determine a structure for your database. If you are moving from a manual database, that structure may already exist.

The method of determining a structure for a database is similar to that for a manual system. However, in addition to identifying what fields you require, you will also need to determine field lengths. Be generous, if your system will allow, so that you are not continually changing field lengths. Separate fields will need to be given field names or tags to which the computer attributes information that is being input.

For example, you might decide to call the 'organisation' field ORG and the 'address' field AD. Where a field consists of several lines, such as an address, it is advisable to split these into separate fields, particularly if you want to search on individual elements of an address field or to print them out as separate lines. An address field might thus be divided into two or three separate fields for building/house name, street, community/ neighbourhood /village and fields for town county and postcode. So your tags might look like AD1, AD2, COM, TOWN, COUNTY, PCODE. In some cases, a record might contain several repetitions of the same field types, e.g. names and addresses. Every time these occur you could give them the same tags, but this may cause problems when conducting searches or printing out information.

For example, if you wanted to see or print out a list of organisations in, for example, the 'Parkside' neighbourhood, the computer would also throw up records for, say, the secretary or second contact who lives in the 'Parkside' neighbourhood, if they all have the tag COM. The simplest way to get round this is to add to the root name for tags a separate number for each occurrence within your record structure, e.g. COM1, COM2, COM3, etc.

There are occasions when it is advantageous to use the same tag for several fields:

1. You might, for instance, want to have several fields for subject headings. By giving them the same tag name, this will ensure that, in whatever subject field a particular term is entered, the computer will throw up the record when a search is made under that term. It will also ensure that when records are printed out, the record will appear under each subject heading.
2. When you want to link files in a relational database system, this can be achieved by having a field with the same tag name appearing in each file, e.g. subject,

name of organisation, client, etc. A search on that particular field will throw up records in all the files bearing the term entered.

It is advisable to have the database form in the same order as the form that you send out to organisations for their information, as this makes it easier for the person inputting the information. One of the advantages of having a computerized database is that it enables you to automate administrative processes.

For example, update letters can be customized and printed out as you determine. If you want to update continually throughout the year rather than in one session, then you will need to build into your record structure, field that enables you to do this, e.g. date record last updated, or added, AGM date. Dates must be entered in a consistent form. Alternatively, you could update a section of the file each month by the initial letter of each organisation, e.g. Month 1: A-B, Month 2: C-D, and so on. When sending out update letters or forms for information about new organisations, if people's names and addresses are going to be recorded, make sure that you conform to the requirements of the Data Protection legislation by including a statement to the effect that: 'This information is being stored on a computer. Please indicate any items of information that you do not wish to be recorded.' Details of the Data Protection legislation are available in a series of free booklets from the Office of the Data Protection Registrar, Springfield House, Water Lane, Wilmslow, Cheshire SK9 5AX.

When sending out an update letter, it is also helpful to send a printout of the organisation's current entry as a prompt for any changes that may have occurred and to save the secretary time in filling out information that has not changed.

All database systems will have some facility for sorting records according to predetermined parameters, e.g., alphabetical by subject, then alphabetical by name of organisation or by date, record number, place, etc. You can usually set up several sort codes for different purposes. The sorting facility is used mainly as a prerequisite when printing out records as a list. For this, you will use another element of a database management system, the report function. (name may vary according to DBMS used). This enables you to print out information in a predetermined format, using all or a selection of fields in a record. Again, more than one report can be set up to take account of different uses. In some integrated packages, the database records have to be fed into the word-processing package for printing out.

UTILITY OF DATABASES

You will need to consider potential uses of the database. Is it a tool for the staff alone to use in answering enquiries? Are you going to allow public access to records either as printouts or through an online terminal? Are you going to make the database available to other organisations through exchange of disks or even sell it to interested parties? The answers to some of these questions will have implications for the kind of hardware and software you choose, and therefore need to be decided at an early stage.

For example, if you decide to allow public access via an online terminal, apart from the additional terminal and keyboard you will also need a multi-user version of the software which must be capable of building in safeguards to prevent the public either accidentally or deliberately erasing or changing records. Some database systems offer run-time versions of the software at a substantially lower price. These allow the system to be run but cannot be used to make changes to it or to develop something new. A good example of this is Help for Health's Cardbox database, where subscribers receive a run-time version of Cardbox and regular updated disks of the database. Obviously, a run-time database will not be as up to date as a true online system but could be a cheaper and safer alternative. You can also get video-text packages which run on PCs and enable you to marry attractive graphics with a database. A good example is the TAP (Training Access Point) terminals located in various public buildings to dispense information on training opportunities, both local and national. The national information was obtained through online links with remote databases. The local information database is maintained on a master PC and then transferred by disk at regular intervals to the PCs driving the TAP terminals. The advantages of this kind of system are:

— Eye-catching graphics can be used to overcome the public's reluctance to use computers;

— Protects your main database;

— Less staff-intensive.

The disadvantages are:

— The information is not as up to date as a true online system;

— A PC is occupied just in running that function;

— It can become monopolized by young people who think it's the latest arcade game;

— If a printout facility is offered, this can easily be abused, requiring frequent staff attention;

— There is a need to make the equipment inaccessible or tamper-proof

PC-based video-text terminals have been used to good effect by some commercial organisations to provide local information and advertising. You find these most often in the windows of tourist information centres, the Computer display being operated by a touch-sensitive keypad attached to the window. Another variation is to get a graphics package, such as Animator, which has the facility of enabling the computer to rotate a series of frames of information at set intervals. Several years ago COIC (Careers & Occupational Information

Centre) offered a video-text package called Rotaview which enabled users to create colourful graphics and link these with a database.

The system was used to prepare an off-line database, to access or download frames of information from Prestel, or to operate as a mini-viewdata service with users dialling into the database for information. Another attraction of Rotaview was its carousel feature that enabled up to 80 frames of information to be scrolled at intervals. A particular use for this type of feature in an information centre would be to display essential information to the public out of hours. Rotaview was marketed at schools, colleges and public libraries but, alas, was developed only for the BBC micro and is no longer available. However, with infinitely more powerful PCs around today, there must be similar software on the market, if you are interested in this way of displaying information.

Seven

DEVELOPMENT OF COMPUTER

In Japan manufacturing productivity is currently growing at the rate of 4.1 per cent a year. France and Germany have manufacturing productivity growth rates of 4.9 and 5.0 per cent a year, respectively. Meanwhile, in the US the rate of growth in manufacturing productivity has fallen sharply. From 1969 to 1973, output per man-hour increased at a compound rate of 2.9 per cent. From 1973 to 1979, the gains dwindled to 1.6 per cent a year. In the US the programme to reverse this productivity pattern significantly must rely on the continued development of advanced technology and its application.

Perhaps the most important element in this reliance on innovation is increased factory automation and a growing use of computers and microprocessor technology in manufacturing. Today, we are on the technological and sociological edge of a dramatic increase in the use of computers in our factories. This will have a profound impact on the nation's productivity growth in the next decade. Within the next 10 to 15 years, four evolutionary trends will meet on the factory floor:

(1) The increasing power and simplification of computers,

(2) A widespread appreciation of the practicality of computerized manufacturing and robotic applications,

(3) A new realization of the impact of computers on people and of people on computers, and

(4) A growing awareness of the urgent need for manufacturing innovation in our society.

Many computerized factory systems exist today as islands of automation. The immediate task of the scientific and technical communities is to use the increased power and simplicity of computers to link these elements into an integrated system. Making use of low-cost computer hardware to perform more and more jobs will make such an integrated factory system economically viable. Some difficult technical problems remain. We must develop generally accepted, standardized interfaces between computerized design engineering and computerized manufacturing, between individual machines and machining centres, and between computers and the people using them.

We must also refine the present state of application technology and reduce the cost of the factory automated system through the increased use of computers. Advances in factory automation are dependent on advances in computer technology.

EFFECT ON SOCIETY

The easier it is for people to use computers, the broader will be their applications in manufacturing. We are moving away from the aeroplane cockpit approach, with rows of complex devices, to create simple computer tools. Compare the ease with which we use personal computers today with the way we approached computers in the early 1970s.

The same changes will occur in manufacturing. The computer has three language levels: machine language, programming language such as COBOL or FORTRAN, and user interface or problem-oriented language. Designing the computer so that it can be quickly used by someone familiar with a problem-but not with computers-is the most difficult of all programming tasks. Today, computers "converse" with users in pictures, in ladder diagrams, or in the secretarial language of word processing.

Voice recognition systems will free workers' hands to perform other tasks and make it even easier for them to use computers. In the future, when research efforts begin to pay off in systems with some understanding of natural language and with "common sense," communications with computers may become as simple as talking to a three-year-old. An important area of concern in manufacturing is worker safety. Automated systems and robots must be able to work side by side with humans.

Major developments are taking place in the sensor area, particularly with proximity detectors and ultrasonic sensors, to make robots more suitable for inclusion in existing factories. The application of automated systems in manufacturing will have several major affects on the people involved in production. It will make our jobs more interesting and challenging; it will enhance job security; and it will multiply the productivity increases. Workers today are looking for greater job satisfaction through greater involvement and increased sophistication. New technologies provide this added dimension to the workplace. For instance, draftsmen use CAD today to perform work that was normally performed by engineers just five or ten years ago. Engineers, in turn, are freed to delve into even more technically sophisticated areas. As a peripheral advantage, the critical need for technical manpower is partially satisfied.

To manage technological change, we must manage our human resources better. For example, we must commit ourselves to ensuring that none of our workers is laid off because of technological changes, as long as they are willing to be retrained and accept new job assignments. Our experience at Westinghouse has been that employees displaced by robots normally move up to better, more challenging work. We should also rethink who can do what job.

For instance, there is a tremendous potential for productivity improvement if the person who knows the machine better than anyone else also has the skills to programme the machine while it is working on another job and the skills to debug the programmes on the spot. In Westinghouse, by putting the programmes for people first, we expect to multiply the productivity improvements that are gained through technology and capital investments. With participative management, for instance, employees welcome advanced technology because they feel in charge of it. Only when these programmes are in place will we emerge from the showcase and token automation phase that manufacturing is presently in.

Today, scientific work in the application of computers to factory automation is in the embryonic stage. We are on the verge of seeing the cost of NC, CNC, and robotics become low enough for these systems to be economically justifiable for many more applications. The cost will continue to decline as application problems are resolved and the computer becomes an understood and respected partner in the manufacturing environment.

First came numerically controlled machine tools, which performed their operations automatically according to coded instructions on paper or Mylar tape. Then came computer-aided manufacturing, or CAD/ CAM, which replaced the drafting board with the CRT screen and the numerical control tape with the computer. The new systems integrate all these elements. They consist of computer-controlled machining centres that sculpt complicated metal parts at high speed and with great reliability, robots that handle the parts, and remotely guided carts that deliver materials.

The components are linked by electronic controls that dictate what will happen at each stage of the manufacturing sequence, even automatically replacing worn-out or broken drill bits and other implements. Measured against some of the machinery they replace, flexible manufacturing systems seem expensive. A full-scale system, encompassing computer controls, five or more machining centres, and the accompanying transfer robots, can cost $25 million. Even a rudimentary system built around a single machine tool-say, a computer-controlled turning centre-might cost about $3,25,000, while a conventional numerically-controlled turning tool would cost only about $175,000.

But the direct comparison is a poor guide to the economies flexible automation offers, even taking into account the phenomenal productivity gains and asset utilization rates that come with virtually unmanned round-the-clock operation. Because an FMS can be instantly reprogrammed to make new, parts or products, a single system can replace several different conventional machining lines, yielding huge savings in capital investment and plant size. Flexible automation's greatest potential for radical change lies in its capacity to manufacture goods cheaply in small volumes.

Since the era, of Henry Ford, the unchallenged low cost production system has been Detroit-style "hard" automation that stamps out look alike parts in huge volume. There is little flexibility in hard automation's transfer lines, which get their name from the transfer of the product being worked on via a conveyor from one metalworking machine to another. But such mass, production is shrinking in importance compared with "batch production" in lots of anywhere from several thousand to one. Seventy-five percent of all machined parts today are produced in batches of 50 or fewer.

Many assembled products too, ranging from aeroplanes and tractors to office desks and large computers, are made in batches. Even such stalwarts of inflexible mass I production as the automakers are developing systems to produce more low-volume models for small market segments. In the past, batch manufacturing required machines dedicated to a single task. These machines had to be either rebuilt or replaced at the time of product change. Flexible manufacturing brings a degree of diversity to manufacturing never before available. Different products can be made on the same line at will. General Electric, for instance, uses flexible automation to make 2000 different versions of its basic electric metre at its Somersworth, New Hampshire, plant with total output of more than I million metres a year. The strategic implications for the manufacturer are truly staggering. Under hard automation the greatest economies were realized only at the most massive scales. But flexible automation makes similar economies available at a wide range of scales.

A flexible automation system can turn out a small batch or even a single copy of a product as efficiently as a production line designed to turn out a million identical items. Enthusiasts of flexible automation refer to this capability as "economy of scope." Economy of scope shatters the tenets of conventional manufacturing. There is no long trip down the learning curve on the factory floor, thanks to the unprecedented precision the system brings to each step of the manufacturing process, from machining to inspection.

The manufacturer will be able to meet a far greater array of market needs, including quick-changing ones seeing the needs of markets the company is not in now. He can keep up with changing fashions in the marketplace-or set them himself by updating his product or launching a new one. He has many more options for building a new plant: FMS frees manufacturers from the tyranny of large-scale investments in hard automation, allowing construction of smaller plants closer to markets. Flexible manufacturing is the ultimate entrepreneurial system: it will allow fast-thinking manufacturers to move swiftly into brand-new fields and to leave them just as swiftly if need beat the expense of less agile older producers. As the new tools come increasingly into use, "some companies will find themselves blind-sided by competitors they never imagined existed," says Joseph D. Romano, a vice-president at A. T. Kearney Inc., management consultants. Flexible manufacturing systems were developed in the US more than ten years ago by Cincinnati Milacron, Kearney and Trecker, and White Consolidated. The US remains a world leader in the technology: the major machine tool builders are being joined by new suppliers with great financial resources and technical abilities, such as GE, Westinghouse, and Bendix.

The joint venture will bring together GM's considerable capabilities in design and software and Fanuc's expertise in building and applying robot systems. GMF plans to start building products next year. However, most of the action in flexible automation is now in Japan, and both American and European manufacturers will soon start feeling the pressure. Like many other manufacturing technologies conceived in the US—among them numerically-controlled machine tools and industrial robots-the FMS was greeted with a yawn by US manufacturers.

The Japanese have become the implementers par excellence of this new type of factory automation, not because they are great technical innovators, which they admit they are not, but because they have moved fast in putting the new systems into their factories. Once again, the path to success in a new manufacturing method leads through those Japanese factories set up as spotless little towns with flower beds and tree-lined streets. A visitor to Japan these days finds the new manufacturing system turning out parts for machine tools in Nagoya, electric motors near Mount Fuji, diesel cylinder blocks in Niigata, and many other products elsewhere. In most cases these plants run on three shifts. During the day skeleton crews work with the machines. At night the robots and the machines work alone.

In Fanuc Ltd's cavernous, bumblebee-yellow buildings in a pipe forest near Mount Fuji, automatic machining centres and robots typically toil unattended through the night, with only subdued blue warning lights flashing as unmanned delivery carts move like ghostly messengers through the eerie semi-darkness. This plant, one of two in the Fuji complex, makes parts for robots and machine tools (which are assembled manually, however). The machining operation, occupying 54,000 square feet, is supervised at night by a single controller, who watches the machines on close-circuit TV. If something goes wrong, he can shut down that particular part of the operation and reroute the work around it. Some Americans think that Fanuc's Fuji complex is just a showcase. Some showcase.

The total cost of the plant was about $32 million, including the cost of 30 machining cells, which consist of computer-controlled machine tools loaded and unloaded by robots, along with materials-handling robots, monitors, and a programmable controller to orchestrate the operation. Fanuc estimates that it probably would have needed ten times the capital investment for the same output with conventional manufacturing. It also would have needed ten times its labour force of about 100. In this plant one employee supervised ten machining cells; the others act as maintenance men and perform assembly. All in all, the plant is about five times as productive as its conventional counterpart would be.

Across the street, 60 machining cells and 101 robots toil in a big two-storey facility automatically machining parts and assembling them into 10,000 electric motors a month. There is nothing else like it in the world. Men perform maintenance functions here in the daytime. The robots work through the night, in silence marred only by hydraulic sighs and the sibilance of those automatic carts. The first floor of the plant contains the machining cells and 52 robots. Machining is carried out on about 900 types and sizes of motor parts, in lots ranging from 20 to 1000 units.

Machined parts are temporarily stored in an automatic warehouse; they are automatically retrieved when they are scheduled for assembly on the second floor. Yamazaki Machinery Works Ltd. operates a flexible automation plant near Nagoya that makes parts of computerized numerically-controlled lathes and machining centres; the latter combine several metalworking machines and incorporate automatic tool changers. In the daytime 12 workers man the $20 million plant. At night only a lone watchman with a flashlight is on duty while the machines keep on working. A conventional machining system with similar production volume, according to Yamazaki, would require 215 workers and nearly four times as many machines, and would take three months to turn out the parts the new plant makes in three days.

The company estimates that over five years of operation its plant will produce after-tax profits of $12 million, compared with $800,000 for a conventional plant that size. Yamazaki is now transferring this technology to its machine-tool-making plant in Florence, Kentucky, bad news for Yamazaki's American competitors. But the most astonishing Japanese automated factory will be started up by Yamazaki about 20 miles from its headquarters near Nagoya. This will be what Tsunehiko "Tony" Yamazaki, the personable senior executive managing director, describes as his company's twenty- first-century factory. The new plant's 65 computer-controlled machine tools and 34 robots will be linked via a fibre-optic cable with the computerized design centre back in headquarters. From there the flexible factory can be directed to manufacture the required types of parts-as well as to make the tools and fixtures to produce the parts-by entering into the computer's memory names of various machine tool models scheduled to be produced and pressing a few buttons to get production going.

The Yamazaki plant will be the world's first automated factory to be run by telephone from corporate headquarters. The plant will have workmen, to be sure: 215 men helping produce what would take 2500 in a conventional factory. At maximum capacity the plant will be able to turn out about $230 million of machine tools a year. But production is so organized that sales can be reduced to $80 million a year, if need be, without laying off workers.

The Yamazaki plant illustrates yet another aspect of economy of scope: with flexible automation, a manufacturer can economically shrink production capacity to match lower market demand. Though Japanese machine tool makers are the most ambitious installer's of flexible automation, they are by no means alone. FMS is spreading throughout Japanese manufacturing, with Panasonic, Mitsubishi, and other consumer and industrial goods producers installing the new systems. So far, nothing even remotely comparable is happening in manufacturing in the US or anywhere else in the world. Disturbingly, all of US industry can boast only about 30 flexible manufacturing systems in place; in Japan one large industrial company, Toyoda Machine Tool Co., has more than 30. Frets David Nitzan, director of industrial robotics at the research and consulting firm SRI International, "We are facing another sputnik-a Japanese sputnik."

The growing Japanese lead underscores frequently heard charges that US managers are too remote from technical disciplines to appreciate the potential of such new technologies, and too engrossed with short-run financial results to invest in them. More often than not, machine tool makers report, executives of US manufacturing companies look at something like a flexible machining system only in relation to the narrowly defined functions of the conventional tools it might replace-not for its potential to provide a different, and far more efficient, Organisation of the manufacturing process. "Cost accounting is a very poor language to communicate new ideas in," observes Paul R. Haas, vice president of Kearney and Trecker's special-products division. One consequence of this myopia is apparent in the ageing of the US machine tool stock. "Many

American factories are barrier reefs—one old, tired technology piled on top of another," says James A. Baker, the executive vice president in charge of GE's drive to develop automated systems. "Even when they build new factories, Americans tend to use the same old machines, shipping them from the old plant to the new plant."

Uncle Sam is the Methuselah of machine tools: more than 34 per cent of US tools are 20 or more years old, the highest proportion in any major industrialized nation. Even England is better off; only 24 per cent of its machine tools are similarly ancient. In Japan, only 18 per cent of machine tools are 20 or more years old; 61 per cent are less than 10 years old, against 31 per cent in the US. In fact, fewer than 4 per cent of machine tools installed in the US are numerically controlled—though the concept has been commercially available for a quarter of a century.

The art of managing the factory is no less antiquated. Fixated by the short run, managers have pursued all sorts of piecemeal efforts to hold down costs without stopping to map out systematic ways of organizing the factory floor for more efficiency. Too often production procedures in US factories appear to be little more than accretions of ad hoc solutions to problems ranging from space shortages to union-dictated work rules.

Managers focus obsessively on chipping away at direct labour costs rather than exploring better ways to organize the work-force—or investigating the extent to which new technologies are making direct labour costs less important. Japanese corporate leaders, by contrast, tend to be sympathetic with technical disciplines—a far greater proportion of them are engineers—and they have more freedom to incur short-run costs in pursuit of long-run strategic objectives. Perhaps most important of all, a technology like flexible automation is a logical extension of a manufacturing philosophy that views the production of goods as a seamless activity that starts with product design and ends with support in the field a philosophy, as the Japanese put it, of "making the goods flow like water."

"Japanese management takes a holistic view of manufacturing," says James F. Lardner, a Deere and Co. vice-president who supervised a major restructuring of the farm equipment company's manufacturing operation. "They apply logic and common sense to their problems rather than laboratory investigations and discounted cash-flow calculations."

Even before they began to adopt flexible automation, Japanese plants typically employed far fewer people for a given output than American and European plants. The 'Japanese were able to do that by reorganizing production-including the placement of machine tools on factory floors. In the US and Europe, machine tools are usually grouped by type, and parts are directed to them as required.

The Japanese instead place different types of machines together so that each given part can be processed in one place. Much quicker than anyone else, too, the Japanese have taken to such important concepts as "group technology "the grouping of similar parts into families for easier manufacture and better inventory control. Like so many other Japanese manufacturing methods, group technology isn't new; the idea evolved in the 1920s in Germany. A central element of the Japanese manufacturing philosophy is the famous just-in-time concept, the system in which materials and components are delivered as required on the shop floor, not accumulated and stored for future.

Since one-third of factory space is usually employed for storage, the savings are substantial. But there is much more. By reducing inventories to the lowest level at which operations can be sustained, the Japanese force their manufacturing Organisations to deal with problems previously hidden. For example, the Japanese already practised preventive maintenance on their machine tools to a degree unknown in the US. Just-in-time has forced them to do even better, because the flow through of products requires every single machine to function perfectly all the time. Yet the Japanese do nothing that Westerners can't—if they only decide to do it. Marvellously efficient factories using the latest automated equipment exist in the US and Europe, towering like islands of excellence in a sea of stagnation and yielding remarkable benefits to their owners. Take Deere and Co's giant new tractor assembly plant in Waterloo, Iowa, which need not take a back seat to any plant in any industry anywhere.

In the past few years, Deere has restructured the Waterloo complex from a gigantic, somewhat chaotic job shop into a world-class producer. Deere has poured $500 million into the complex, $150 million of it into the 2.1 million square foot assembly plant. Chassis and engines received from sister plants are joined with tractor cabs and bodies made at Waterloo into gleaming mechanical behemoths.

Almost all the materials handling at Waterloo is under computer control. Each part or sub-assembly-engine, transmission, wheels, and so on is automatically assigned to a specific customized tractor ordered by a dealer; it is retrieved from storage and delivered automatically to the assembly line just when it is needed. Putting just-in-time to work, Deere has cut inventory in some areas by as much as 50 per cent, saving millions of dollars.

Flexible automation allows Deere to build a tractor at least twice as fast as before. And it has given the company a new agility: Deere can now successfully compete not only against other big manufacturers but also against "short-liners" that make only one farm implement in higher volumes.

What's more, the company is right now bidding on a defence contract worth hundreds of millions of dollars. If it wins the contract, it will start making bulldozers, graders, and other heavy construction equipment on its new flexible automation lines-putting new pressure on such established manufacturers as Caterpillar and Case.

But if manufacturers think that they have to pour hundreds of millions of dollars into flexible automation to reap its rewards, they are mistaken. They can begin by acquiring smaller machining centres to modernize portions of their operations. Clifford R. Meyer, president and chief operating officer of Cincinnati Milacron, recalls his delight at recently visiting one of his client companies, which occupies a garage at the end of an alley in a Los Angeles suburb. Inside, a father-son entrepreneurial team mans $450,000 worth of the latest computerized machine tools—successfully competing as a parts supplier to aerospace companies against much larger firms with older production equipment. Furthermore, not all plants that install flexible automation have to be started from scratch. GE, Ford, and GM are among the manufacturers that have successfully revitalized old plants by installing new machinery.

Like many other American companies, GE had for years lagged in automating its own factories. Some old GE plants, James Baker notes, "make Santa's workshop look like the factory of the future." Today, however, GE can boast notable successes in converting some of its old plants into what it calls "factories with a future"—a marketing slogan the company uses to impress upon potential customers that they need not build entirely new factories of the future. Its metre plant in New Hampshire, modernized at a cost of $25 million, is the epitome of an antiquated multi-storey mill building.

Another ancient GE plant, the Erie, Pennsylvania, locomotive facility, is being transformed with a $300 million investment into an ultramodern automated factory—inside if not on the outside. Building a batch of locomotive frames formerly took about 70 skilled machine operators 16 days; the newly automated factory will turn out these frames in a day—untouched by human hands. The displaced workers are being retrained for other, more sophisticated jobs. As a general matter, in fact, flexible automation threatens employment less than might be supposed.

The US faces a shortage of skilled machinists for the rest of the decade, and automation of assembly, where semi-skilled jobs predominate, will proceed much more slowly than automation of machining. GM is also advancing. Last October it installed its first flexible automation system, an Italian-built Comau system with three machining centres, at the Chevrolet Gear and Axle Division in Detroit. Almost immediately GM discovered just how valuable the new manufacturing flexibility can be. When an outside supplier failed to deliver a front-axle component up to quality standards, GM brought the job in-house. It designed and built the tooling for the component on the FMS in ten weeks—a job that normally would have taken up to a year. As yet, such examples are rare exceptions.

The majority of US manufacturers either do not yet grasp the significance of the new technologies, or do not want to invest while money is still expensive and future markets uncertain. Moreover, there are still no turnkey flexible manufacturing systems available. Installation of big systems requires skilled people not all companies may have on hand. But as GE's Baker argues: "We're running out of time and excuses." The price of delay may be disaster, as the experience of some American machine tool manufacturers show.

During the 1970s the Japanese became the world's first mass producers of computerized machine tools. While the majority of US machine tool makers—small companies, in the main—plodded along using old technologies and methods, the Japanese redesigned their tools for easier manufacturing with flexible automation and equipped these tools with advanced yet simple electronic controls that the humblest job shop could understand. Then, in the late 1970s, the Japanese caught the US tool-makers napping in the midst of the capacity crunch, with delivery times stretched up to two years.

The Japanese, to be sure, played an additional trump. Their machine tool industry had been mobilized by Japan's Ministry of International Trade and Industry (MITI) into a cartel and bolstered with millions of dollars in government funds, as shown in documents obtained in Japan by lawyers for Houdaille Industries, the US machine tool maker that has asked President Reagan to deny investment tax credits for some Japanese tools sold in the US. This double-barrelled assault left US makers of numerically controlled machining centres and lathes in shambles. By the end of last year the agile Japanese had captured more than 50 per cent of the US market for those machines.

The US machine tool industry has lost its erstwhile position as the world's leading producer to competitors not only in Japan but also in Europe and even Taiwan. The National Academy of Engineering, usually not given to alarmist statements, in a report soon to be issued calls the Japanese and other foreign inroads "a very threatening development that could seriously endanger the future economic security of the American industry."

The reason for concern is simple: the machine tool industry is central to the growth of all manufacturing. There will be more Japanese surprises in the years to come. This October, for instance, the Japanese government's mechanical engineering laboratory near Tokyo will unveil a small prototype factory of the future where novel, laser-equipped machine tools will take production automation a big step beyond where it is now. The new machines will perform metalworking processes now done separately—such as turning, drilling, and milling—all at once, cutting batch production time of metal parts in half and reducing the number of production processes by 60 per cent.

MODERNISATION SKILLS

The technology exists in many parts of the world to achieve significant advances in many areas of factory automation. Many

European countries are working on numerical control, process planning, and group technology approaches rather than the graphics approach emphasized in the US. They are well advanced in integrating CAD and CAM systems. In Japan, major advances are achieved through the efforts of the Ministry of International Trade and Industry (MITI). For example, hundreds of millions of dollars and some of the finest minds in business and universities are being applied to the task of developing a fifth-generation computer.

Another example is a $60 million government funded project to develop a flexible machine system. This system will use high-energy lasers to manufacture small batches of machined parts with assembly line efficiency. The project involves more than 500 engineers from 20 Japanese companies, and it could revolutionize much of manufacturing. MITI makes use of Japan's homogeneity and Organisational milieu. It would be inappropriate for the US to adopt the same methods to foster manufacturing innovation.

However, we must recognize the urgency of the problem. The US is one of the few major industrialized nations in the world without a significant coordinated industry–government-university programme directed at improving manufacturing technology. We need a national strategy for productivity improvement that brings together government, business, labour, and academia in a cooperative, rather than adversary, relationship. We will have to remove many of the disincentives to innovation and find new ways to capitalize on our diversity and our proven creative and inventive abilities. Just as MITI capitalizes on Japan's homogeneity, we must find new ways to foster, encourage, and channel our innovative diversity.

At present, Westinghouse is working with the National Science Foundation and the universities of Rhode Island, Florida, and Wisconsin on technology development programmes. Along with the Robotics Institute of

Carnegie-Mellon University, we are developing "seeing," "feeling," and "thinking" robotic systems for several of our factories. We are also very interested in the Air Force's ICAM programme to coordinate sophisticated design and manufacturing techniques now used by industry on a piecemeal basis. This programme attempts to integrate design, analysis, fabrication, materials handling, and inspection and to develop hardware and software demonstration manufacturing cells in selected aerospace plants.

EVOLUTION OF COMPUTER

The technological changes in the computer field during the past several years clearly equal any technological change that has occurred in our society over the past 100 years. The introduction of high-performance, low-cost microprocessor and storage technology has dramatically improved and enhanced the functions and capabilities of computer software and hardware. With today's increased power, manufacturing computer systems can be made more adaptable to the manufacturing environment, thus cutting systems engineering costs and time per installation. The cost of computer hardware itself has been steadily declining. This trend will continue with the introduction of new technologies such as very high-performance microprocessor chips based on very large scale integration (VLSI). On the other hand, the cost of the human and software resources for systems engineering and programming has gone up. In 1955, 85 per cent of the total cost for processing information was hardware; it is estimated that by 1985 hardware will account for only 15 per cent of this total cost. To improve manufacturing productivity we must reverse this trend by optimizing the use of our human resources and taking advantage of the increased computer power and reduced hardware costs. The need to make increased use of available computer power is heightened by the decline in available technical manpower.

The National Science Foundation found the annual growth rate of scientific and R&D personnel between 1954 and 1969 to be 5.9 per cent. Some 5,56,000 employees were involved in technical work in 1969, but the number fell to 5,17,000 in 1973 and then grew to only 6,10,000 by 1980s—a rate of only 2.8 per cent annually. The increased capability of computer technology and broad availability of application software packages, including the expanded use of problem-oriented languages and database software, can reduce the cost of programme development and maintenance by a factor of 10.

The graphics capabilities of engineering computers can link integrated design and drafting systems to manufacturing systems. With these "user-friendly" approaches to provide a bridge between man and computer, the user can interact directly with the manufacturing system without traditional interfaces and jargon-heavy manuals. Computers and the applications they support can now be considered for functions or activities that only recently were impractical. The opportunity for the application of this technology, for all practical purposes, is unlimited.

STUDY OF ROBOT

The problems encountered in trying to integrate advanced computer concepts, new manufacturing technologies, and the shop floor are clearly evident in the evolution of robotics. Robots are classified according to the way we provide them with information and the amount of self-adaptability they possess. The most *comprehensive categorization of robots* is provided by Japan's Industrial Robot Association:

— Manual manipulators are worked by an operator.

— A fixed-sequence robot has a manipulator that repetitively performs successive steps of an operation according to a predetermined sequence which cannot be easily changed.

— A variable-sequence robot is similar to the fixed sequence robot except that the set information can be easily changed.

— Playback robot reproduces, from its computer memory, operations that were originally executed under human control.

— An NC robot is a manipulator, whose tasks are programmed by using numerical control tapes or cards.

— An intelligent robot, using sensory perception, detects changes in the work environment and proceeds accordingly, using its decision-making capability.

Industry today is focusing on the development of NC and intelligent robots. There has been a growth in the number of firms that manufacture and sell such robots which is reminiscent of the proliferation of minicomputer companies in the 1960s.

Basically, robots are microprocessor-controlled mechanical devices that perform a function or provide an intelligent interface between machines and processes. They can be intelligent enough to make on-the-spot manufacturing "decisions." But for robots to become practical, we must reduce their size, mechanical complexity, and installed cost—primarily through the expanded use of computer and control technology.

Robots can duplicate human manipulative skills with accuracy and precision. Their flexibility and versatility, as opposed to hard automation, make robots ideally suited to the kinds of small batch jobs that constitute the bulk of industry's manufacturing activity. Today, robots, are freeing people from jobs that present serious health hazards, are mundane, or are highly repetitive. In most cases their use is justified for non-economic reasons. In the US, industry has been slow to adopt robotics. This reluctance appears to be due primarily to the large initial investment and the general availability of relatively inexpensive manual labour. Why install a $100,000 or \$150,000 robot to perform a \$25,000-a-year job? Also, the majority of today's robots are monsters: bulky, unwieldy mixtures of hydraulic and mechanical contraptions with a machine tool heritage.

This situation is changing. Robots are becoming more streamlined, and, when they are manufactured in large quantities, will rapidly decline in cost. Many technologically innovative firms are entering the business. Equally important, system engineering, which represents as much as two-thirds-of the cost of a robotic application, is being greatly reduced. It is not difficult to imagine that in a short time the cost of a typical robotic system will be paid back in one or two years.

In the next decade the cost of a robot is likely to be down to \$10,000 to \$20,000, while skilled labour costs might easily be \$25 or \$30 an hour. When this economic threshold is reached, there will be a virtual flood of robotic applications. When this happens, robots will play an important part in the totally integrated factory of the future. Most of our plants will have a direct numerical control supervisory computer that coordinates the activities of several NC and CNC machines or hardwired machining centres and robots and connects all the machines into a system. The robot interface will handle the transfer of material and, with newly developed sensory capabilities, will also perform the in-line inspection of parts. For the near future, however, our factories will be some particular mix of machines, robots, and people that makes the most economic sense. Eventually in a decade or so-robots will fill a void in the supply of skilled labour. There has been a shift in the labour force from blue-collar to white-collar workers and from production jobs to service jobs. Currently, about two-thirds of our work force and 85 per cent of all college graduates are employed in a service- related activity.

The total service-oriented labour force is expected to increase by 20 per cent in the 1980s—to about 85 million people. This shift to a service economy, coupled with the slowdown in the growth of the US population, suggests that many businesses will find factory labour in short supply: this is already the case in Sweden today. At a leading university in Japan, there is a robot with human-type hands and legs, TV-camera eyes, artificial ears and mouth, and touch and joint sensing. These technologies are combined to provide the robot with some of the capabilities of a two or three year old child.

For example, when ordered to fetch an item in the room, the robot looks around the room and finds the article, walks to it, picks it up, and brings it back. If the robot does not understand a command, it speaks up. Such robots are essentially showcase examples: they are not appropriate for the majority of industrial applications. In fact, a universal person-like robot would make little sense except for very limited, specialized applications. At present, the major applications for robotics are in arc welding and material transfer. In the future, the major application will be in assembly. Artificial intelligence is a worthwhile goal,, but for the moment industry has more than enough applications for "dumb" robots. Employment of robots in these applications is actually limited by the extensive engineering required to put them to use. It is estimated that by 1990 two-thirds of the robots sold to industry will be off-the-shelf, modular units rather than specially designed systems. Looking at automated systems generally, by the end of the decade, 30 per cent of our systems will consist of hard automation, about 20 per cent will be adaptive control, and the remaining 50 per cent will be systems of a universal programmable nature. Respondents to a study conducted by the Society of Manufacturing Engineers ranked the technical and performance barriers constraining the rapid utilization of robots in US industry.

The leading technical barriers were mechanical manipulation, vision systems, tactile systems, sensory systems, programming, and control systems. The primary performance barriers included accuracy, speed, and the ratio of capacity to size. Robots will have a much greater impact in manufacturing when their total installed cost is reduced and they are more easily programmed. As with CAD and CAM systems, the most difficult thing about putting in a robot is interfacing it with the factory—both the machines and the people.

Research today is focused on the development of (1) equipment that will make greater use of computer technology to cut the cost of systems engineering and power electronics, improve servo-motor technology, and rapidly move from hydraulics to electric; and (2) sensors that will enable robots to perform more reliably and with greater precision and adaptability.

A good *overview of sensor types* has been provided by Bejczy:

> The non-visual sensor information is used in controlling the physical contact or near contact of the mechanical arm/hand with objects in the environment. It is obtained from proximity, force-torque, and touch-slip sensors integrated with the mechanical hand. These sensors provide the information needed to perform terminal orientation and dynamic compliance control with fine manipulator motions ...
>
> Terminal orientation and dynamic compliance control are essential and intricate elements of manipulation. Soft and adaptive grasp of objects, gentle load transfer in emplacing objects, assembling or disassembling parts with narrow tolerances, and performing geometrically and dynamically constrained motions (like opening or closing a latch or fitting two parts together) are typical examples of manipulator control problems that challenge both sensor and control engineering ...

Vision systems close the control loop and allow the robot to interact in a dynamic, changing environment. A second use of vision will be for the critical inspection of the batch-produced parts. The position and orientation of the part can be used in advanced automation systems to direct a robot manipulator to pick up the part for an assembly or transfer operation.

The Robotics Laboratory at Westinghouse is working on state- of-the-art applications in many of these areas. Systems and development engineers are working on the integration of controls, tooling, processes, computer, and other elements of the automated factory. Specialists in robotics are concentrating, on developing and applying high-speed vision systems, tactile and force feedback sensors, high-performance electric servo systems, adaptable programmable assembly techniques and computer control, and artificial intelligence.

One of these projects is called APAS (automated programmable assembly system). Funded in part by the National Science Foundation, APAS is a pilot programme in which robots are used to assemble components into the end bells of the Westing-house line of fractional horsepower motors. It is a development project intended to transfer newly developed technology to the factory. The fractional-horse power motors are currently assembled in batches averaging 600, units at a time. There is a 20-second assembly time per motor to put together 30 different parts, and there are 13 changeovers a day to handle 450 different motor styles. The first section of the line puts parts on the motor end bells. To start the 15-second sub-assembly operation, a vision system in conjunction with a five-axis PUMA (programmable universal machine for assembly) robot inspects the end bells to make sure they are the style currently being assembled.

The end bell is then oriented and placed on a pallet. The next step in the assembly is the insertion of the uppermost components: a thrust washer, a bearing cap, and a felt washer. An auto-place robot picks up the parts and loads them onto an anvil. The end bell is moved into the station, and the parts are passed on. At the same time a semi-solid lubricant is injected. At the next station, four screws, a plastic plug, and a contact point are inserted. Following this there is a complicated assembly procedure. In order to assemble all the different styles of end bells, several styles of certain parts are required. More precisely, six styles of mounting rings and three styles of dust caps are needed. Programmable feeders are used to accomplish the feeding and orienting of all these parts. At this station, a PUMA robot picks up a mounting ring, dust cap, and felt washer on an oil finger from the programmable feeder and places them on an insertion device, where they are fitted onto the end bell.

At the final station, a vision system and a PUMA robot are used to perform the final inspection of the end bell, pick it up from the pallet, and remove it from the system. The computer control arid sensory parts of APAS are the most revolutionary elements of the application. A distributed microprocessor system is needed to handle many simultaneous tasks in the short 15-second assembly time, and a visual sensory system is required to provide orientation and feedback.

One master microprocessor controls the entire system. Under its control are three types of smaller microprocessors for Vision control, local process control, and robot path control. All these controllers work in conjunction with the master computer to coordinate the inspection and assembly procedures. The vision system on this project recognizes randomly oriented parts on an assembly line after a multipass learning cycle controlled by an operator. It can also be used to rotate the part to any given angle.

AUTOMATIC MANUFACTURING

Advances in the two primary elements of factory computerization-computer-aided design (CAD) and computer-aided manufacturing (CAM) will create a new industrial revolution. By integrating design with manufacturing, we can not only turn out new product designs much faster, but also programme the computer to make sure the designs provide quality and reliability as well as the lowest possible manufacturing costs. CAD/CAM is the integrated use of advanced computer technology in engineering and manufacturing. It is a common database of part and product geometry and related information which makes it easier to translate a creative idea into a final product at a reduced cost. With CAD, a user can define a part shape, analyse stresses and other factors, check mechanical actions, and automatically produce engineering drawings from a graphics terminal. When CAD is combined with the CAM system, the user can also manipulate non-graphic data such as bills of material, shop information, and cost factors.

The end result is greater design flexibility and what is referred to as designing to cost. The CAD functions can be grouped in four categories: design and geometric modelling, engineering analysis, kinematics, and drafting.

— In *design and geometric* modelling, the designer describes the shape of a structure with a geometric model constructed graphically on a cathode-ray tube. The computer converts this picture into a mathematical model, which is stored in the computer database for later use. Many other design functions depend heavily on the model. It can, for example, be used to create a finite-element model for stress analysis, serve as input for automated drafting to make a drawing, or be used to create numerical control tapes for the factory.

— After the geometric model has been created, the engineer can easily calculate such things as weight, volume, surface area, moment of inertia, or centre of gravity of a part. But the most powerful method of analysing a structure is probably finite-element analysis. In this technique, the structure is broken down into a network of simple elements that the computer uses to determine stresses, deflections, and other structural characteristics. The designer can see how the structure will behave before it is built and can modify it without building costly physical models and prototypes. This procedure can be expanded to a complete systems model, and the operation of a product can be simulated.

— With *computer kinematics*, the user can examine the effects of moving parts on other parts of the structure or design and can analyse more complex mechanisms.

— Finally, the CAD system automatically drafts drawings for use in manufacturing.

Computer-aided design is a good example of the transition of expensive, state-of-the-art computer technology to a commercial, economically justifiable system. Recent advances in CAD technology have increased the productivity and effectiveness of design engineering groups. Such systems will be even more common in the next five to ten years. Manufacturing groups can draw on the geometric and numerically coded description produced by CAD to create numerical control tapes, which allow direct computer control of shop machines, determine process plans and scheduling, instruct robots, computerize testing, and in general improve the management of plant operations.

Computer-aided manufacturing has five main functions: tool design, machine control, process and materials planning, robotics, and factory management.

— Manufacturing engineering and *tool design* deals with the machines and fixtures needed to make a new product. In effect, the set of machines, tooling, and fixtures is a new product, and all the techniques of CAD are used in fashioning it. The CAD techniques are then used to simulate plant operation and the integration of machines and materials handling.

— *Machine automation* consists of a chain of increasingly sophisticated control techniques. At the lower end of the spectrum are fixed automation with relays or cams and programmable controllers, where relays have been replaced by electronics. Moving up the spectrum, numerical control (NC) refers to controlling a machine with pre-recorded, numerically coded information to fabricate a part. In this case, the machine is hardwired and not readily reprogrammed. In computer numerical control (CNC) the machine is directly controlled by a minicomputer, which stores the machining instructions as software that is relatively easy to reprogram. Because of the computer control, CNC has the advantages of much higher storage capability and increased flexibility. Virtually all numerical control is computer-based, yet only ten years ago CNC was an expensive exception.

— *Process planning* considers the detailed sequence of production steps from start to finish. The process plan describes the state of the workpiece at each work station. An important element in process planning is group technology, in which similar parts are organized into families to allow standardized fabrication steps; this permits significant savings by avoiding duplicate tooling and systems engineering. Most automated process-planning systems use a retrieval technique based on part families and existing databases for standard tool in and fabrication processes. Materials planning or manufacturing resource planning is concerned with the precise flow and timing of manpower, materials, and processes; it is a detailed look at how everything comes together. The ultimate goal is to have continuous use of all production equipment, no bottlenecks, and a minimum inventory.

— Because they are widely applicable, *robots* have a distinct advantage over specialized, highly engineered manufacturing systems. The economic advantage of a mass-produced, readily adaptable robot over a one-of-a-kind system with a great deal of engineering content is obvious. Robots are now being used to perform materials -handling functions in CAM systems. They can select and position tools and workpieces for NC or CNC tools, operate tools such as drills and welders, or perform test and inspection functions. Through visual or tactile sensors, the robot can manipulate objects. Through its computer intelligence, it can inspect the object and provide the machine with corrective feedback or actually reprogram the machine or change the tooling.

— *Factory management* coordinates the operations of an entire plant. Factory management systems tie together individual machine tools, test stations, robots, and materials-handling systems into manufacturing cells and the cells into an integrated whole. An integrated CAM system of this sort is usually hierarchical, with Microprocessors handling specific machining functions or robot operation, middle-level computers controlling the operation and work scheduling of one or more manufacturing cells, and a large central computer controlling the overall system.

Reliability is greatly improved by structuring the control system correctly. Local, distributed control (with defined responsibilities) reports up to a supervisory control that, in turn, is linked to a managerial computer. This parallels the structure of the typical industrial Organisation.

Ultimately, the digital output from the CAD computer will be simply plugged into the CAM system to reprogram the plant's manufacturing computers. In such an integrated system, the databases will be organized in a way that avoids redundancy and reformating of information. And any change in one part of the system will automatically revise dependent or related information in other parts of the system. Bridging the CAM and CAD systems will be one of our major jobs in the future.

A fundamental difference that has to be reconciled is that CAD makes use of a pictorial, graphics oriented computer database, while CAM involves a great deal of text-oriented information. In other words, we need to find a way for the computer doing the drawing to speak the same language as the computer directing the manufacturing plant. Layering is one way to link these systems.

Layering is a particular technique for structuring the CAD and CAM databases. It enables various people to input data without losing control of the overall design and manufacturing process. Equally important, it enables shop people to see information that is meaningful to them without having to sort through and understand the rest of the information that is normally included in a drawing. To do this, all information is organized in an arrangement resembling layers, or slices, inside the database.

The engineer or users in other departments of an Organisation can provide pertinent information or examine any or all layers of information according to their particular needs. As an example, a printed circuit board may have 250 to 300 layers of information.

A manufacturing engineer inputs layers of information that deal with fabrication and assembly. In turn, machine operators concerned with the details of the drilling and cutting configuration may access layers dealing with this part of the drawing. Other layers provide information pertinent to the needs of the purchasing department or component assemblers. Another major effort to integrate computer systems is an Air Force programme called ICAM (integrated computer-aided manufacturing).

This is a practical attempt to greatly shorten the time span for the implementation of compatible and standardized computer-manufacturing techniques and to provide a unified direction for industry.

The ICAM programme provides seed money for the establishment, within private industry, of modular subsystems designed to computerize and tie together various phases of design, fabrication, and distribution processes and their associated management hierarchy. As appropriate, these mutually compatible modules will be combined to demonstrate a comprehensive control and management package capable of continual adjustment as production needs and the state of the art change.

The ICAM programme is divided into five major parts.

— Defining the manufacturing architecture. This permits a concentration on problems of generic scope and wide applicability as the basis for later projects in integration, support, and application systems and demonstrations.

— Developing integration methodology. This activity provides a bridge between industry and ICAM for the transfer of ICAM technology for the integrated factory of tomorrow. The projects addressed include establishing factory simulation techniques, ICAM implementation techniques, configuration management, modelling tools, software integration simulation, automated systems engineering methodology, and various system analysis and design capabilities.

— Establishing support systems. This is concerned with the portion of the ICAM system involving computer operations, including both software and hardware and both operational and managerial aspects of computerized manufacturing.

— Establishing application systems. This includes such items as manufacturing cost and design guides, the design manufacturing interface, manufacturing standards, group technology concepts, and scheduling and process planning. Under an ICAM contract, the National Bureau of Standards considered standards in computer communications, languages, and networks to identify potential conflicts within an integrated manufacturing environment. Other areas of concern include robotics, prototype integrated production cells, integrated materials-handling and storage systems, and integrated manufacturing control and material management.

— Demonstrating the ICAM programme. The ultimate goal in ICAM is the use of totally integrated manufacturing systems by industry in the completely automated factory.

Eight

NEW GENERATION

A major development in the short- to medium-term future will be a substantial growth in multimedia telecommunications in one form or another. In this way, multimedia information networks will correspond more closely to the experience of communication and information interchange in everyday life, with a flexible mix of sound, picture, and words. The exact nature of this experience, and its effect on information networking in general, remains to be determined.

A major pre-occupation of national governments and regional associations is to build the infrastructure for these developments as quickly as possible (and preferably faster than their competitors). In the United States, for example, in an attempt to accelerate the extension of fibre optic cabling to the home and workplace on a massive scale, the Federal Communications Commission (FCC) decided in 1992 to allow the regional Bell holding companies to transmit video programmes over their telecommunications lines, in competition with the cable television companies. This attempt to create a communications infrastructure for the next decade is seen by many Americans as crucial to the continued economic dominance of the United States.

The FCC, for example, places much weight on multimedia devices such as high definition television (HDTV), a digital two-way interactive television set, as the vehicle for this information expansion, aiming at a 15-year deadline for complete HDTV conversion. The technological futurism is the subject of much political controversy and manoeuvring at the current time; it seems likely though, that the new presidential administration will lead out more liberalization to support competition-led innovation and technological advances in communications media, and there is a clear emphasis on the use of technology for economic development in the shaping of public policy statements.

One more immediate result is likely to be the more rapid development of the National Research and Education Network, sponsored for many years by Vice-President Al Gore. The building of NREN services will bring into focus many of the issues on information access and price, and of the emerging adjustments in role of all those involved in the information cycle. In Europe, a similar urgency has fuelled a number of projects sponsored by the European Commission to work towards the creation of integrated broadband communications systems by the latter part of this decade.

The RACE programme, funding research into advanced communications technology and services, indicates by its name some of its underlying objectives, and the ESPRIT programme has initiated the IDOMENEUS project to coordinate and improve European efforts in the development of new information environments in the area of interacting open media.

Adopting lessons learned from the French experience in the establishment of the Minitel system, the EC has initiated a number of pilot projects in fields ranging from banking and finance, manufacturing, transport, distribution, and health care;

other projects include distributed multimedia publishing, and a broadband network based library for the petroleum and chemical industries, with a 1995 target date for the introduction of commercial services.

PRESENT GENERATION

The current supercomputers are only at the threshold of what computer designers think can be achieved; the next generation of advanced supercomputers will make today's machines look like handheld calculators. "We have problems that would take 500 to 1,000 hours to solve [on today's supercomputers]," says David Nowak, division leader for computational physics at Lawrence Livermore National Laboratory, where a cluster of seven supercomputers-known as "Octopus" are used for nuclear weapons research.

Before the end of the twentieth century, computer scientists had developed machines that not only crunched numbers at high speed but also exhibited artificial intelligence— computers that think and reason somewhat like human beings and that could understand information conveyed by sight, speech, and motion. The question was, which nation's scientists would get there first? The Japanese had announced a two-pronged plan to build advanced computer technologies.

One project was the $100 million, eight-year National Superspeed Computer project, which aimed at producing machines 1000 times faster than the earlier Cray-1 supercomputer built by Cray Research of Minneapolis. The other, the $500 million, 10-year Fifth Generation Computer project, was focusing on artificial intelligence. Both were being countered by American efforts, including a Pentagon request for up to $1 billion over the next five years for superspeed and artificial intelligence (AI) technologies. Although behind, Great Britain and France have also launched national supercomputer projects. The great danger for the losers in the race-and the opportunity for the winners was that whoever built the next generation of computers would have a huge technological and commercial advantage: these computers would be used for computer and microelectronics design-to build even smarter, and more powerful machines. They won't be self-replicating machines, but they would be close. "It takes you a long time to catch up," says computer scientist Rai Reddy of Carnegie-Mellon University, one of the top US computer-research centres. "In some of these areas, that is the difference between a first-rate power and a second-rate power-from an economic point of view and from a security point of view. " The losers in the race will fall farther and farther behind. The leading edge of computer science is still a black art; there are no fixed laws, and the field is highly experimental. That is what worries US scientists about Japan's approach—some success is inevitable. "Because the field is experimental, [the Japanese] will come out with something," says Dertouzos. "It may not be what they wanted, but they'll come up with new architectures, new insights, and new design techniques.

To build the computers that dominated the 1990s, both Japan and the US are depending on the onrushing technological advances in microelectronics. Japan's Fifth Generation project will use faster, denser circuitry to create a new class of superintelligent computers.

The 24 projects in the Fifth Generation concentrate on artificial intelligence, a goal of American computer scientists for more than a quarter century. "We are trying to catch up to you, and not the other way around," says Tokyo University Professor Tohru Moto-oka, who organized the project for the Japanese government. Although the Japanese are highly regarded as superb engineers, Japanese computer scientists have often been faulted for failing to develop innovative computer software.

In a break with tradition, Kazuhiro Fuchi, the Fifth Generation project director, has deliberately assembled a young team: "The question was who would adapt most easily to this research," says Fuchi. "Young people have fewer fixed ideas." The project, head-quartered in a downtown Tokyo skyscraper, will focus on computer architectures, software, and the symbolic logic necessary to build thinking computers. In the US, meanwhile, three huge new programmes in electronics and computing are getting under way.

MICROELECTRONICS AND COMPUTER TECHNOLOGY CORP. (MCC)

Last year William Norris, the founder and chairman of Control Data Corp., convened a meeting of top computer and semi-conductor industry executives at the Grenelefe Golf and Tennis resort in Oriando, Florida, to discuss setting up a hugh research cooperative.

The companies agreed to form a non-profit joint venture so that they could pool their resources and share the cost of doing long-range research. Twelve major US corporations joined the new Organisation, including Honeywell, Motorola, RCA, and Control Data. MCC is a bold departure from the way research is usually done in American universities and corporations, and it will probe the limits of the nation's antitrust laws. The venture partly follows the Japanese model: the companies will donate scientists and researchers to MCC, loaning them for up to four years.

The corporate co-owners will also put up the money to fund MCC' research, in return for the rights to use the results. Whether the scheme will work is an open question. The 12 companies in MCC are competitors in fast-moving, high-technology markets, and ordinarily they jealously guard any technological edge they gain. In fact, many top US "firms—Cray Research, Texas Instruments, Intel, and others—chose to stay out of MCC. "That's not our style," says John A. Rollwagen, chairman of Cray Research, an 11-year old company proud of its entrepreneurial creed. "We don't want to participate." The biggest market force of all-IBM-reportedly stayed out of MCC because it feared anti-trust action against it if it joined. So far, however, the creation of MCC has not provoked any such suits.

San Francisco anti-trust lawyer Joseph M. Alioto did write to the chief executives of the companies that were about to form MCC: "In my opinion, your contemplated conduct is an unequivocal combination in violation of the anti-trust laws of the United States." But the threat did not deter MCC's co-owners and, for the time being at least, the Justice Department has allowed the MCC plan to stand. To run the new corporation, MCC's directors chose retired Admiral Bobby Ray Inman, former director of the National Security Agency and former deputy director of the CIA. Inman is widely respected for his managerial abilities and is an adept politician besides. "The day they picked Bob Inman to head MCC," says George W. Keywork II, Ronald Reagan's top science adviser, "any concern about its success diminished in my mind."

Over the past five months, Inman orchestrated a competition among 57 cities for the MCC headquarters; the winner was Austin, Texas, after private donors, the state, and universities put together a generous package of incentives.

The consortium will have a budget of about $75 million a year and a staff of 250. Its first projects include programmes in semi-conductor packaging and interconnect technology, advanced software engineering and computer-aided design and manufacturing (CAD/CAM) for the electronics and computer industries. Most ambitious is a 10-year programme aimed at breakthroughs in computer architecture, software, and artificial intelligence.

MCC will own the licenses and patents to the technologies; the manufacturing and marketing will be left to the companies that sponsor the projects. MCC win give them a competitive edge on the market—they will have exclusive rights for three years before the research is published and other firms are allowed to buy licenses.

SEMICONDUCTOR RESEARCH CORP. (SRC)

Over the past few years Japan has captured a vital segment of the world semi-conductor industry, the market for so-called RAM (random access memory) chips, a technology invented in America.

Japan now supplies 70 per cent of all 64K RAMS sold, and it appears that, as the next generation of memory chips, the 256K RAMS, is being readied for market, the Japanese semi-conductor companies are threatening to take a big share of sales. It makes for a grim reminder: a decade ago, before an all-out government project to build up the industry, Japanese semi-conductor firms lagged far behind American and European chip companies. Last year 13 US chip manufacturers and computer companies banded together to form a non-profit research consortium, the semi-conductor Research Corp. (SRC), to share the spiralling costs of advanced research and development. SRC's founders include Control Data Corp., Digital Equipment Corp., Hewlett-Packard, MM, Intel and Motorola. Unlike MCC, however, SRC, which is headquartered at Research Triangle Park in North Carolina, does not carry out its own research. Instead, it sponsors research at universities. SRC is spending $12 million this year and has allocated $30 million next year. The goal: "To assure long-term survival in the market," says Larry Sumney, SRC's executive director.

DARPA: The Pentagon's Defence Advanced Research Projects Agency (DARPA) is, more than any other single agency in the world, responsible for the shape of advanced computer science today-and for many technologies now in widespread commercial use.

Over the past 20 years, DARPA has poured half a billion dollars into computer research, in the process virtually creating the science of artificial intelligence.

The first supercomputer, built in 1964, was a DARPA project. Computer time sharing, a fundamental advance, came out of work sponsored by DARPA. So did packet-switched networks, the workhorses of today's telecommunications data networks. And computer graphics—now used on desktop computers and video-arcade cockpits—is a DARPA-sponsored invention. DARPA's next priority is a push for advanced supercomputing and artificial intelligence technologies that may cost as much as $1 billion.

DARPA plans to do everything the Japanese have set out to accomplish-and more. Earlier this year DARPA proposed a "Strategic Computing and Survivability" project, which it hopes will lead to a variety of new machines. "We want some architectures that are good for building semantic memories, memories that can hold knowledge," says DARPA's computer director, Robert Kahn. "Other kinds of systems are good for logic processing. We want architectures that can do very rapid signal processing [and] structures that can handle very, very large amounts of data in communications."

Once they are in place, these technologies will make possible an astonishing new breed of weapons and military hardware. Smart robot weapons—drone aircraft, unmanned submarines, and land vehicles—that combine artificial intelligence and high-powered computing can be sent off to do jobs that now involve human risk. "This is a very sexy area to the military, because you can imagine all kinds of neat, interesting things you could send off on their own little missions around the world or even in local combat," says Kahn.

The Pentagon will also use the technologies to create artificial intelligence machines that can be used as battlefield advisers and superintelligent computers to coordinate complex weapons systems.

An intelligent missile guidance system would have to bring together different technologies-real-time signal processing, numerical calculations, and symbolic processing, all at unimaginably high speeds—in order to make decisions and give advice to human commanders. While the national security needs are driving supercomputing technologies, there is a growing market for commercial spinoffs.

The same technology that can be used to simulate an anti-tank missile smashing into a heavily armoured tank can also be put to work on less martial arts. In Los Angeles, Digital Productions, Inc., is using a Cray supercomputer to produce television commercials for Mattel, rock video segments for Turner Broadcasting, and special effects for Lorimar's "Star Fighter" space epic, which will be released next summer. Instead of shooting a commercial the conventional way—a costly photo session—Digital can create the "pictures" it needs in detail so precise that it's impossible to distinguish between the supercomputer's graphic image and the real photo. For inanimate objects, that is: the Cray-1/S can simulate a car right down to the glint of sunlight on the windshield, but not a human being. As the commercial market grows, the Japanese believe that their strategy of building supercomputers for the general-purpose market will pay off.

During the past year Fujitsu, Hitachi, and NEC have each announced supercomputers faster than the most powerful American machines now on the market, the Cray X-MP and Control Data's Cyber 205. Moreover, all three are designed to use the standard Fortran language, lifted from ordinary mainframe computers, but at much faster speeds. "Although supercomputers right now are mainly being used by specialists," says Takamitsu Tsuchimoto, Fujitsu Corp.'s development manager for supercomputers, "we believe that in the next 5 to 10 years they will be used by a lot of ordinary people, so we wanted to design a machine [they could use] without making great efforts." Designing a supercomputer is no easy task. America's premier supercomputer designer is Seymour Cray, 57, who designed the Cray-I and the basic architecture of Control Data's Cyber 205.

The Cray x-mp contains a dense pack of 240,000 silicon chips arranged to shorten the distances the electrical signals must travel, thereby decreasing the time it takes to perform an operation; the new Cray x-mPs will run at 400 million operations per second. And Japan's superspeed goal is 10 billion operations per second. (An Apple IIe computer contains 31 chips and can execute 500,000 operations per second.) Because of the cost of supercomputers—Cray's x-mp will sell for $11 million—the market will remain limited. Japan's Fifth Generation project, however, plans to build "super personal computers."

These huge increases in raw computer power are just the first step. The most profound changes brought by the new technologies will be the development of reasoning computers that will use superspeed symbol manipulation to simulate human thought. "We are about to see the next explosion, which is the application of computers to reasoning," says Stanford University computer scientist Edward A. Feigenbaum, who is a founder of two artificial intelligence companies, Teknowledge Inc. and IntelliGenetics, and co-author of a new book on Japan's challenge, *The Fifth Generation*.

Limited forms of artificial intelligence are already enjoying commercial success. Digital Equipment Corp. the nation's second largest computer firm, uses an AI programme called X-CON to make custom designs of its, computer systems.

Using a set of more than 2500 rules programmed into the system, X-CON examines a customer's specifications, determines whether all the necessary components are included, then draws a set of diagrams showing the proper spatial relationships among the components. Scientists studying artificial intelligence have been aided by advances in computer hardware, too. In 1980 a group left MIT's Artificial Intelligence Laboratory to found a company, Symbolics Inc., of Cambridge, Mass., to build computers specially designed to run LISP, a language used to develop artificial intelligence programmes;

and Xerox has begun to sell similar computers. Symbolics has been selling its machines, the Symbolics 3600, to a broad range of university and industrial research labs.

The company has another customer as well: the managers running Japan's Fifth Generation project have bought 10 Symbolics machines—and they have 15 more on order. In exploring the brave new world of artificial intelligence, computer scientists are concentrating on several important problems. "Knowledge engineers" are building the so-called expert systems that can mimic human expertise in a narrowly defined area. The firm of Teknowledge, in Palo Alto, California, for example, has built an expert system for the French Elf Acquitaine Oil Co., a system that will give advice on one of the industry's most costly technical problems—what to do when a drill bit gets stuck thousands of feet below the earth's surface.

To build the system, Teknowledge engineers interviewed Elf Acquitaine's top troubleshooter, Jacques-Marie Courte, and then programmed his answers into a computer. The programme is, in effect, a computer replica of Courtes expertise; the computer will ask the drilling-rig foremen questions, just as Courte would.

Once it gathers the information it needs, the computer will make recommendations by drawing images on the screen and giving suggestions on how to retrieve the bit. Because daily drilling costs are high, Elf Acquitaine may well recover the program's development costs the first time it is used successfully. General Electric is building a software programme that will provide expert advice on repairing locomotives.

The Pentagon would like to build artificial intelligence programmes that could serve as a pilot's assistant in the cockpit. Stockbrokers and insurance agents may also soon get help: "Some people are beginning to see a gold mine in [building artificial intelligence programmes for] financial services," says

Patrick Henry Winston, chairman Of MIT's Artificial Intelligence Laboratory. And savvy software designers for the personal computer industry are beginning to look at artificial intelligence as the next big wave that will sweep the market. Despite the successes, there are problems ahead as researchers attempt to move beyond the building of narrowly defined "experts." "There's a lot of hard stuff out there we just don't have the answers to," says Roger Schank, director of Yale University's Artificial Intelligence Laboratory. Computers have always been maddeningly literal machines, subject to the absolute tyranny of the binary codes they use to do their calculations. (The switches are either on or off, simulating ones or zeros, nothing else.)

That literal-mindedness can transform a problem that would seem trivial to humans into a nightmare. Consider the simple statement "Mary had a little lamb."

For a computer to translate the text into another language—a function scientists are now trying to develop-it would have to sort through what is, by one count, 28 possible interpretations (Mary owned the lamb, Mary ate the lamb, Mary had sexual relations with the lamb, Mary gave birth to the lamb, and on and on).

The kind of understanding humans experience as a "flash of recognition" is also difficult to instill in a computer. The statement "Ronald Reagan is president" carries a number of immediate meanings to a flesh-and-blood American, but a computer would have to rummage through its silicon memory chips in search of dozens of facts—what the word "president" means and biographical facts and details that tell who Ronald Reagan is. And the biggest challenge of all—teaching computers to learn to acquire knowledge on their own—is nowhere near being solved.

Mankind has long been enchanted—and frightened—by the prospect of creating machines that think. "I don't see any limitations to artificial intelligence," says Nobel laureate Herbert Simon, professor of computer science and psychology at Carnegie-Mellon. "All the mechanisms for human intelligence are present. Already machines can think just like people—in a limited sense. Man isn't unique in that respect." Moreover, many machines may soon possess sight, touch, hearing (in the form of voice recognition), and speech, thus imitating humans' sensory capacities along with their intellectual ones. But whole areas of the human thought process—volition, emotion, the creative uses of error—still lie well outside a computer's experience. And scientists doing research on artificial intelligence are far from their ultimate goal—a computer-based analog of the human brain.

Still, the race to build superintelligent computers—Japan's challenge to the US—will almost certainly push the technology to new levels. "While at the beginning I was angry at Japan because I felt that they were swiping our best ideas, really, on second thoughts, I couldn't blame them," says Dertouzos of MIT. "And I don't blame them today at all. I think they're doing exactly what they should." And the Japanese challenge has at last spurred the US into action. "If we really wake up, I'm very optimistic," says Dertouzos. "We could beat the daylights out of them." No one knows where the competition will ultimately lead.

VISUAL PHONE

The existence of a technology, however, does not always guarantee its success. It is worth remembering that the Picture Phone was launched by AT&T at the World Fair of 1964. This was a first example of the videophone; in 1992, almost 30 years later, British Telecom launched a low-cost consumer videophone, which can be connected directly to the normal telephone socket; this has yet to make any substantial impact in the market-place.

A development which is likely to be of more significance, though, is the planned availability of audio-visual workstations based on the standard PC. Such a device will make the full integration of data and voice networks within the workplace a reality, and will open up a wider market for multimedia information services. Many of these exist in prototype form within the existing infrastructure, Multimedia electronic mail, for example, is now possible even in a limited text-based medium such as the Internet.

The multimedia information mail extensions (MIME) to the standard Internet mail protocol permit the inclusion of still images, group III fax images, compressed full motion video, and multipart playback sequences of mixed information types. Desktop videoconferencing, the ability to conduct meetings with a geographically remote person each at their own desk, has been established as a feasible communications technology, although so far finding only a niche rather than global market. The experience of this videoconferencing raises some key issues since it is clear that, as with most other information technologies, the technical issues are much less important than the social and organisational ones. Technical advances which have made the idea more feasible include the emergence of high bandwidth ISDN networks, video compression techniques, accepted standards, and powerful graphics-based workstations.

The degree of success in implementing this technology is determined rather by the extent to which it supports the way people work. This involves an understanding of the communication patterns between individuals and groups, and of their information needs and the uses to which information is put; of the way in which contacts with individuals and formal information sources assists the innovative process, and an appreciation of the interdependent balance between formal and informal communication.

Technology has often been used to support formal communication patterns, rather than the informal, and sometimes is intended to replace rather than support either.

Multimedia communications systems cannot replace face-to-face contact any more than the telephone (although a similar argument was made against the telephone at its introduction). Thoughtful design may, however, be used to appropriately support existing human communication patterns. The Cruiser Videoconferencing System designed by Bellcore, for example, uses the metaphor of a 'virtual hallway'—floor plan representing the groups of participants—along which a user may navigate, 'cruise' and interact with others in the process. For multimedia multi-user systems such as this, the interface must include some form of social protocol which enables decisions about the nature and style of desired interaction to be taken.

Many issues which will effectively delineate the shape of multimedia networking in the future remain as yet poorly understood and little explored. In some areas, however, there is more immediacy on concerns which are being raised by the substantial growth in network traffic and users, and the potential changes which can be perceived in the near future.

DIGITAL INFORMATION

The growing importance of digital information and its availability over networks which demand physical and technical access create barriers, which are increasingly insurmountable for many groups of potential users. While there is no space here to discuss the social role of information systems, there are a few examples of attempts to bypass these barriers in recent developments. The Cleveland FreeNet experiment, for example, is an attempt to increase access to Internet information to a wider audience. The vast amounts of information, and the worldwide nature of the Internet, also bring to the fore issues of intellectual property and copyright surrounding electronic information. Much of the information held on this network is freely available, as long as it is used for non-commercial purposes.

The subsequent use of information taken from computer networks cannot be controlled, however, once it has been down-loaded from the original source, it may appear in many diverse guises, be re-partitioned, amalgamated, and reused, taking on a life of its own. Print-based copyright legislation does not translate easily to the electronic environment. A feature of the legislation to found the advanced NREN network requires the provision of charging mechanisms to be put in place, suggesting a trend towards more control over use and users.

The characteristics which make electronic information difficult to define and 'freeze' at one point in time also raise questions on the nature of bibliographic control and mapping of information resources. Electronic documents and journals which can be updated immediately, replacing the original copy, are difficult to define as a bibliographic entity which has a declared and fixed content, which can be cited for later use. The identification of various versions of a logical item becomes a complex (or perhaps impossible) task, as does preservation and archiving as a permanent record of publication. These problems also rebound upon ethical issues involving the quality and accuracy of the document content. Electronic documents may easily be changed, or 'falsified' with changing intellectual fashions in a way that print being fixed at one point in time cannot.

The unstructured and fluid process which allows a network user to make information available without recourse to formal publisher or refereeing process has great advantages in stimulating an innovative and creative dialectic, but at the cost of a large proportion of false, misleading, or incorrect information which must also be sorted through and rejected. Whilst the research community, in which much of this type of activity takes place at present, has evolved a fairly sophisticated code of practical ethics for network cooperation and exchange, it is doubtful that this will survive a broadening of access and the introduction of commercial interests.

The increasing ubiquity of the networking environment is raising questions about the relative responsibilities of those involved in the information transfer process. Traditionally, this process could be described in terms of five distinct roles: the information provider, who provides the original source material; the information host, who makes it available and accessible, 'publishing' in a print or electronic sense; the carrier or distributor, providing a channel for the information; the intermediary, traditional role for the information service professional; and the information user, the final consumer. This model has long ago begun to break down when applied to the networking environment.

The information provider is often the same as the information host; the common carrier, the telecommunications authority, provides information services in its own right; and the information user more often wishes to find and retrieve the information directly. Many people have begun to question the role of the publisher in the electronic environment: since the author can mount a document in its electronic form on a local computer (quite probably the format it was originally composed in) and the network facilities act as a distribution mechanism, it becomes harder to gauge the function of a traditional publisher. Although this is clearly not appropriate for many types of publishing, in some areas it is possible to see advantages.

For instance, the traditional venue for reporting research has been through a journal article in some summarized form; network publishing would allow a reader to optionally download original datasets and intermediate analysis, as well as authorial commentary, directly and without a time delay. The development of network publishing is likely to involve identifiable changing characteristics compared to the traditional activities:

— A growth in publishing, with one source file producing a variety of complementary versions in different formats;

— Multimedia electronic publishing, integrating database, text, image, sound, and video;
— The use of information tools to skim, navigate, browse, and utilize these new media products;
— Selective and customized publishing targeted at particular groups;
— A linking with the concept of a 'virtual library', where access and materials are not constrained by physical barriers.

The information user and the information provider potentially have a much more direct and interactive relationship, possibly to the extent that the information exchange cannot be completely separated from the communication process itself. This raises a question mark against the received role of the intermediary, as a filter between originator, distributor, and recipient.

As the growth in electronic information begins to demand a new philosophy of the nature and function of information, so also the activities of the information professional must, in this setting at least, change and adapt. No longer able to adopt a custodial role in managing and organizing the materials and artefacts of information, the librarian or information scientist must adopt a facilitating strategy, aimed at creating the conditions for a user to negotiate the information sphere successfully. The librarian no longer 'owns' the information since it is not contained within any physical institution; and it cannot therefore be 'managed' in the same physical sense. The information user, however, must still negotiate an information space, now across a digital network as much as before within the confines of a library, in order to achieve a successful resolution to the information transfer process.

Nine

NEXT GENERATION

Japan's Next Generation project will use faster, denser circuitry to create a new class of super intelligent computers. The 24 projects in the Fifth Generation concentrate on artificial intelligence, a goal of American computer scientists for more than a quarter century. Although the Japanese are highly regarded as superb engineers, Japanese computer scientists have often been faulted' for failing to develop innovative computer software. But in a break with tradition, Kazuhiro Fuchi, the Fifth Generation project director, has deliberately assembled a young team: "The question was who would adapt most easily to this research," says Fuchi. "Young people have fewer fixed ideas." The project, headquartered in a downtown Tokyo skyscraper, will focus on computer architectures, software, and the symbolic logic necessary to build thinking computers. The Microelectronics and Computer Technology Corp (MCC). Last year William Norris, the founder and chairman of Control Data Corp., convened a meeting of top computer and semi-conductor industry executives at the Grenelefe Golf and Tennis resort in Orlando, Florida, to discuss setting up a huge research cooperative.

The companies agreed to form a non-profit joint venture so that they could pool their resources and share the cost of doing long-range research. Twelve major US corporations joined the new organisation, including Honeywell, Motorola, RCA, and Control Data. MCC is a bold departure from the way research is usually done in American universities and corporations, and it will probe the limits of the nation's anti-trust laws. The venture partly follows the Japanese model: the companies will donate scientists and researchers to MCC, loaning them for up to four years. The corporate co-owners will also put up the money to fund MCC' research, in return for the rights to use the results. Whether the scheme will work is an open question. The 12 companies in MCC are competitors in fast-moving, high-technology markets, and ordinarily they jealously guard any technological edge they gain. In fact, many top US firms—Cray Research, Texas Instruments, Intel, and others—chose to stay out of MCC.

MCC's directors chose retired Admiral Bobby Ray Inman, former director of the National Security Agency and former deputy director of the CIA. Inman is widely respected for his managerial abilities and is an adept politician besides. Over the past five months, Inman orchestrated a competition among 57 cities for the MCC headquarters; the winner was Austin, Texas, after private donors, the state, and universities put together a generous package of incentives.

The consortium will have a budget of about $ 75 million a year and a staff of 250. Its first projects include programmes in semi-conductor packaging and interconnect technology, advanced software engineering and computer-aided design and manufacturing (CAD/CAM) for the electronics and computer industries. Most ambitious is a 10-year programme aimed at breakthroughs in computer architecture, software, and artificial intelligence.

MCC will own the licenses and patents to the technologies; the manufacturing and marketing will be left to the companies that sponsor the projects. MCC will give them a competitive edge on the market—they will have exclusive rights for three years before the research is published and other firms are allowed to buy licenses.

Over the past three years Japan has captured a vital segment of the world semi-conductor industry, the market for so-called RAM (random access memory) chips, a technology invented in America. Japan now supplies 70 per cent of all 64K RAMS sold, and it appears that, as the next generation of memory chips, the 256K RAMS, is being readied for market, the Japanese semi-conductor companies are threatening to take a big share of sales.

Over the past 20 years, DARPA has poured half a billion dollars into computer research, in the process virtually creating the science of artificial intelligence. The first supercomputer, built in 1964, was a DARPA project. Computer time sharing, a fundamental advance, came out of work sponsored by DARPA. So did packet switched networks, the workhorses of today's telecommunications data networks. And computer graphics—now used on desktop computers and video arcade cockpits—is a DARPA—sponsored invention.

DARPA plans to do everything the Japanese have set out to accomplish and more. Earlier this year DARPA proposed a "Strategic Computing and Survivability" project, which it hopes will lead to a variety of new machines. "Other kinds of systems are good for logic processing. We want architectures that can do very rapid signal processing [and] structures that can handle very, very large amounts of data in communications." Smart robot weapons-drone aircraft, unmanned submarines, and land vehicles-that combine artificial intelligence and high-powered computing can be sent off to do jobs that now involve human risk. "This is a very sexy area to the military, because you can imagine all kinds of neat, interesting things you could send off on their own little missions around the world or even in local comba. The Pentagon will also use the technologies to create artificial intelligence machines that can be used as battlefield advisers and super intelligent computers to coordinate complex weapons systems. An intelligent missile guidance system would have to bring together different technologies-real—time signal processing, numerical calculations, and symbolic processing, all at unimaginably high speeds—in order to make decisions and give advice to human commanders.

The same technology that can be used to simulate an anti-tank missile smashing into a heavily armoured tank can also be put to work on less martial arts. Instead of shooting a commercial the conventional way, a costly photo session, Digital can create the "pictures" it needs in detail so precise that it's impossible to distinguish between the supercomputer's graphic image and the real photo. As the commercial market grows, the Japanese believe that their strategy of building supercomputers for the general-purpose market will pay off. The Cray X-MP contains a dense pack of 240,000 silicon chips arranged to shorten the distances the electrical signals must travel, thereby decreasing the time it takes to perform an operation; the new Cray X-MPs will run at 400 million operations per second. And Japan's super speed goal is 10 billion operations per second. Because of the cost of supercomputers, Cray's X-MP will sell for $ 11 million, the market will remain limited. Japan's Fifth Generation project, however, plans to build "super personal computers." These huge increases in raw computer power are just the first step.

The most profound changes brought by the new technologies will be the development of reasoning computers that will use super speed symbol manipulation to simulate human thought. The Fifth Generation field is not new: scientists in the US have been studying artificial intelligence for more than 25 years, struggling to understand the nature of knowledge and how to represent it in forms adaptable to computer usage.

Digital Equipment Corp the nation's second largest computer firm, uses an Al programme called X-CON to make custom designs of its, computer systems. Using a set of more than 2500 rules programmed into the system, X-CON examines a customer's specifications, determines whether all the necessary components are included, then draws a set of diagrams showing the proper spatial relationships among the components. Scientists studying artificial intelligence have been aided by advances in computer hardware too. In 1980 a group left MIT's Artificial Intelligence Laboratory to found a company, Symbolics Inc., of Cambridge, Mass., to build computers specially designed to run LISP, a language used to develop artificial intelligence programmes; and Xerox has begun to sell similar computers. Symbolics has been selling its machines, the Symbolics 3600, to a broad range of university and industrial research labs. The company has another customer as well: the managers running Japan's Fifth Generation project have bought 10 Symbolics machines and they have 15 more on order in exploring the brave new world of artificial intelligence, computer scientists are concentrating on several important problems. "Knowledge engineers" are building so-called expert systems that can mimic human expertise in a narrowly defined area.

The programme is, in effect, a computer replica of Courtes expertise; the computer will ask the drilling-rig foremen questions, just as Courte would. Once it gathers the information it needs, the computer will make recommendations by drawing images on the screen and giving suggestions on how to retrieve the bit. Because daily drilling costs are high, Elf Acquitaine may well recover the program's development costs the first time it is used successfully. General Electric is building a software programme that will provide expert advice on repairing locomotives.

The Pentagon would like to build artificial intelligence programmes that could serve as a pilot's assistant in the cockpit. Stockbrokers and insurance agents may also soon get help: "Some people are beginning to see a gold mine in financial services," says Patrick Henry Winston, chairman of MIT's Artificial Intelligence Laboratory. And savvy software designers for the personal computer industry are beginning to look at artificial intelligence as the next big wave that will sweep the market. Despite the successes, there are problems ahead as researchers attempt to move beyond the building of narrowly defined "experts."

Computers have always been maddeningly literal machines, subject to the absolute tyranny of the binary codes they use to do their calculations. (The switches are either on or off, simulating ones or zeros, nothing else.) That literal-mindedness can transform a problem that would seem trivial to humans, into a nightmare. Consider the simple statement "Mary had a little lamb." For a computer to translate the text into another language-a function scientists are now trying to develop-it would have to sort through what is, by one count, 28 possible interpretations.

The statement "Ronald Reagan is president" carries a number of immediate meanings to a flesh-and-blood American, but a computer would have to rummage through its silicon memory chips in search of dozens of facts-what the word "president" means and biographical facts and details that tell who Ronald Reagan is. And the biggest challenge of all-teaching computers to learn to acquire knowledge on their own-is nowhere near being solved.

Mankind has long been enchanted and frightened by the prospect of creating machines that think. All the mechanisms for human intelligence are present. Already machines can think just like people, in a limited sense. Man isn't unique in that respect. Moreover, many machines may soon possess sight, touch, hearing (in the form of voice recognition), and speech, thus imitating humans' sensory capacities along with their intellectual ones. But whole areas of the human thought process volition, emotion, the creative uses of error-still lie well outside a computer's experience.

Japanese planners view the computer industry as vital to their nation's economic future and have audaciously made it a national goal to become number one in this industry by the latter half of the 1990s. They aim not only to dominate the traditional forms of the computer industry, but to establish a "knowledge industry" in which knowledge itself will be a saleable commodity like food and oil.

The Japanese plan is bold and dramatically forward looking. It is unlikely to be completely successful in the ten-year period. But to view it therefore as "a lot of smoke," as some American industry leaders have done, is a serious mistake. Even partially realised concepts that are superbly engineered can have great economic value, pre-empt the market, and give the Japanese the dominant position they seek.

GENESIS

Several core research areas are likely to make solid progress within the next decade. Each of these is already being worked on in various countries, and progress does not depend upon the success of Japan's ambitious "Fifth Generation" project. One is low-level vision, based on techniques using parallel hardware and cooperative processing.

Current "connectionist" research in this area differs in its approach from work on two-dimensional pattern recognition by "property lists," and from-the-top-down /Iscene analysis" of three-dimensional scenes. Based on detailed studies of image formation, it is able to extract from the ambient light information about three-dimensional features (such as shape, depth, texture, and surface orientation) which in previous approaches could have been computed only, if at all, by way of high-level knowledge of the expected scene. Some of this work is being done in the context of human psychology and neurophysiology, some in a more technological context.

Dedicated (massively parallel) machines are being designed for this research, and major advances depend upon such hardware. A second area in which we can expect significant progress is robotics. This includes problems of movement control, trajectory planning, and visuomotor coordination (and will take advantage of advances in low-level vision). As in the case of vision, some projects will rely on "artificial" means to ensure success (such as light stripes for automatic welding machines, capable of recognising different sorts of weld joint and guiding the welder accordingly), while others will relate more closely to psycho-physiological theories of motor control and visuomotor coordination in living organisms. Knowledge-based "expert" systems will multiply enormously in the next decade, not least because there is considerable commercial interest in them. Different domains of human expertise may require different approaches to knowledge engineering. In domains less fully covered by an explicit scientific theory. It may be easier to extract knowledge from human experts who are competent but who have not yet achieved the "intuitive" mastery of the domain that top-flight experts enjoy.

The latter give the right answer more often, but cannot easily introspect their reasoning processes, which happen very fast and are not consciously accessible. The former take time to come to a decision, after consciously weighing distinct considerations against each other and verbally identifying areas of unclarity. Domains (such as medical radiology) that depend on the comparison and interpretation of complex visual images are especially difficult to automate, since low-level visual processes are not open to voluntary inspection or control.

Indeed, experts often give highly misleading advice about how they may be carrying out the relevant comparisons. (Eye movement studies show, for instance, that expert radiologists do not scan x-ray photographs in the way they say that they do.) In tandem with the increasing experience of Al-trained knowledge engineers, further psychological studies of the organisation of knowledge in different domains should be useful. Research on expert systems will also focus on the computational architecture required to deal with large, complex knowledge bases.

Current systems are relatively simple and inflexible, and are restricted to very narrow domains. They can be incrementally improved, but only up to a point. Eventually, the interactions between the increasing number of independently added rules become too difficult to control, and the system's reliability and intelligibility are jeopardised. Current systems have no access to higher-level representations of the knowledge domain and their own problem-solving activity. Special problems arise if a system has to work in real-time, where unexpected events can require quick switching from the current activity to some other.

The next ten years will see some general work on powerful (IKBs) architectures (as well as the production of more examples of specific commercially useful systems), including parallel-processing devices. Progress can be expected also in natural language processing, both of individual sentences and of texts. Key issues include syntactic parsing, the integration of syntax with semantics, and the understanding of connected text.

Machine translation could in principle benefit from advances both in single-sentence parsing and in text analysis. Current work on parsing is motivated both by theoretical (linguistic) interests and by the hope of improving the man-machine interface so as to make it possible for non-specialist users to communicate with programmes in (some reasonable subset of) natural language. Where a programme is used for some specific purpose, semantic factors can be more readily used to help in the parsing and disambiguation of queries and instructions input by the user.

Verbal interchanges about lunar geology, or about airline reservations, are already reasonably "natural" because of the exploitation of semantic constraints, and further domain-specific semantics will be developed over the next decade. More generally applicable (theoretical) research will continue into the best point at which to use semantics in parsing: from the beginning of the sentence, or spreading out from the middle, or only after an initial parse of the entire sentence? Text-analysis programmes can already give a precis of most short news stories about specific topics (such as earthquakes, hijackings, and road accidents). But they rely on rigid, pre-programmed schemata, which provide the semantic skeleton of the types of stories concerned.

Some recent research is aimed at enabling a text analysis programme to learn new schemata for itself, to integrate one schema with another so as to understand a story combining both, and to use a given schema to reason analogically in an unfamiliar context. A high degree of success cannot be expected within the next ten years, but our understanding of the relevant problems should be advanced. A variety of educational applications is already receiving attention.

Some are focused on particular curricular subjects, and require both a model of the theory of that subject and a model of the student's knowledge of it (which varies in level and in organisation, from person to person and from time to time). Others are less specific, and aim to use AI-based techniques to improve the pupil's attitude to intelligence in general. There is some evidence that both normal and handicapped students can attain greater self-confidence and intellectual achievement by experience with these specially designed programming environments. Controlled research into the classroom effects of Al-based systems has recently been initiated, and this can be expected to bear fruit within the next decade. An extremely important area, which is increasingly being studied because of recent hardware developments, concerns the computational properties of large parallel systems. At present, we understand very little of the potential and limitations of such systems.

Some of the connectionist work mentioned above suggests that cooperative processing may have some highly surprising properties.

For example, the number of individual processors required to make the "human" range of visual shape discrimination appears to be markedly less than one would naturally assume. Again, making a connectionist system stochastic rather than deterministic improves its chance of finding an optimal solution. The computational properties of parallel machines will not be well understood for a long time, but experience with these new systems in the near future will doubtless lead to some advance.

DIFFERENT TYPES

The impacts of AI on other technologies will include many different examples of applications to individual problems. For example, an olfactory chip is being designed using AI techniques of pattern recognition. Given advances in very large-scale integration (VLSI), instruments and products of many different kinds will come to include chips whose design makes use of Al methods. Any commercial-industrial task that could benefit from even a limited degree of intelligence could in principle be performed better with the help of Al, so the technological applications of Al will be extremely diverse. AI will influence other sciences in their general philosophical approach as well as their specific theoretical content. Indeed, psychology and (to a lesser degree) biology have already been affected by computational ideas. And, contrary to what most people assume, AI has had a humanising effect in psychology. Al's influence will be especially strong in the psychology of vision and language, and, as noted above, it is likely that robotics will engage with the psychophysiology of movement. Psychological research will feed back into Al; for example, insofar as psychologists arrive at a better understanding of the organisation of knowledge, their work may be useful in designing computerised expert systems.

Social impacts will be of various types. First, there will be effects on individuals and institutions brought about by specific applications of Al, such as expert systems for medical diagnosis, legal and financial advice, or educational help. These programmes will not merely provide a service (whose adequacy should be very carefully monitored), but will very likely change the social relations of the profession or institution concerned.

For example, if general practitioners, or nurses, can use an AI programme to aid in various aspects of patient care, the social image of the specialist physician may be profoundly affected. (And legal responsibilities for medical decisions may be assigned in a way very different from today.) Likewise, legal programmes may undermine the status of lawyers, and alter the nature of their work.

In both cases, while the mystique, of the human experts may be lessened, their opportunity for exercising their specifically human powers may be increased. The general public might come to be less dependent on human experts than they are today. Reducing the power of professionals such as doctors, lawyers, and teachers would certainly have advantages. But replacing human professional advice by computer programmes is dangerous to the extent that Al systems in public use are inadequate-and/or ill-understood. Systems that have taken several man-years to develop (and whose original programmers may be retired, or dead) are often very difficult to evaluate or alter, because even computer scientists do not fully understand how they work.

A second type of social impact concerns general social trends brought about by applications of Al and information technology (IT). These include changes in the proportion of the workforce in service and leisure industries, changes in the division of labour and sexual roles, and changes in general lifestyles and patterns of interaction.

But other potential consequences of Al point in the opposite direction. The widespread use of home terminals, for instance, threatens to have an isolating influence even more powerful than that of television. If people are encouraged to work, and to shop, from their sitting-rooms, there may be unfortunate psychological effects in terms of personal stress and loneliness. Community computer centres could offset these effects to some extent, providing a social meeting place outside the confines of the home and nuclear family.

Some writers even predict that commercially available (and highly profitable) Al systems will be heavily used not only in task-oriented ways, but as surrogates for human contact. On this view, the strong tendency to anthropomorphism that most of us share will result in patterns of interaction being skewed away from human beings, and towards quasi-human computer systems (with naturalistic, "voices," and sometimes even "bodies").

MAINTAINING RESEARCH

There will be "more of the same," in that the areas mentioned above will provide perplexing problems for many years to come. Especially hard problems include learning, high-level vision, naive physics, and abstract work in computational logic.

For instance, expert systems are at present unable to "plain their reasoning except by "backwards-chaining": giving a resume of the chain of inferences (rules) that led up to their conclusion. They cannot relate their conclusion to the domain in general, or rely on an overview of the problem to assess the relative theoretical reliability of different hypotheses.

Nor can they monitor and adjust the structure of their own problem-solving, for they have no high-level representation of it. They are unable, too, to integrate different knowledge domains, and to use concepts and patterns of inference taken from one domain to reason (analogically) in another. Nor can current systems explain their conclusions differently to different users, taking account of the specific user's knowledge.

The user can ask for a "deeper" explanation (a more detailed inference resume) but the programme has no user model in terms of which to adjust its explanations to the human's particular range and level of knowledge. For this reason also, the pattern of interaction between user and system is at present very limited. The user cannot offer his own conclusions for comment and criticism, for example, as students can do with human teachers. A special case of human knowledge is "naive physics," one's everyday knowledge of the properties and behaviour of different sorts of physical substances, and the nature of the causal relations between them. This knowledge enters into vision and motor control, and also into natural language.

For example, a language-using programme would have to understand the differences in meaning between verbs such as pour, flow, spill, drop, and the like, if it were to give instructions or understand texts about activities dealing with liquids. Similarly, a robot capable of seeing that a container was just about to spill its contents onto the object below, and of adjusting its movements accordingly, would need some representation of the behaviour of fluids. Very little work has been done on these issues so far, and they are likely to provide a challenge for many years.

"Computer-aided design" is typically thought of as involving the graphical display of precise three dimensional specifications of various products (from machine tools through cars to buildings), taking into account a wide range of values of many parameters. But a recent form of computer-aided design involves suggestion rather than specification, in the sense that the design programme originates novel ideas—ideas that are not merely quantitatively different from previous specifications.

For example, heuristic programmes are already being used to suggest novel experiments (described at the intramolecular level) in genetic engineering, or to help design new sorts of three- dimensional silicon chips. The potential of systems like these should be further explored. The computer modelling of creative thinking will require long-term research, especially with respect to domains whose crucial concepts cannot be so readily defined as the concepts of molecular biology, chip circuitry, or set theory. Machine learning is a pressing problem for the future. If a programme cannot learn for itself, its development is limited by the time and ability of the programmer to provide it with new information and ideas.

MECHANICAL OBJECTIVES

The Japanese Fifth Generation project aims to design and produce computer hardware and software for knowledge engineering in a wide range of applications-including expert systems, natural language understanding by machines, and robotics. To accomplish these goals, the Japanese must improve present computing capabilities dramatically, but they must also make major innovations in existing technology that will enable Fifth Generation computers to support very large knowledge bases, allow very fast associative retrievals, perform logical inference operations as fast as current computers perform arithmetic operations, utilise parallelism in programme structure and hardware to achieve high speed, and develop a machine-user interface that allows significant use of natural speech and images.

All experts systems built by knowledge engineers to date consist of three main parts. First, is the subsystem that "manages" the knowledge base needed for problem solving and understanding. Secondly, is the problem-solving and inference subsystem, which discovers what knowledge is useful and relevant to the problem at hand, and with it constructs, step by step, a line of reasoning leading to the problem solution, the plausible interpretation, or the best hypothesis. Thirdly, are the methods of interaction between human and machine, in modes and languages that are "natural" and comfortable for the user. Ordinary human natural language is often preferred, but the stylised notations of some fields like chemistry are also desirable for specific groups of users. Knowledge—based management, problem solving and inference, and human interaction-these have all been approached in our present expert systems via software innovations, innovations that have pressed traditional von Neumann hardware architectures to their limits.

The Fifth Generation plan organises its work around these three subsystems, but with a critical added dimension: for each component subsystem there is a hardware level and a software level. And between the levels the Japanese designers must define a "language" with which the software and hardware interact. The knowledge in the knowledge base must first be represented in symbolic form, and in memory structures, that can be used efficiently by the problem-solving and inference subsystem. This representation can take many forms.

A typical kind of associative network is the taxonomy, known as "The... is-a (hierarchy)." For example, "The sparrow is-a kind of bird." In this case, both sparrow and bird are objects within the knowledge base. If the knowledge base is informed that "The bird is-a kind of animal that can fly," the knowledge base management system must automatically propagate the deduction that sparrows can fly.

A rule consists of a collection of statements called the "if" part, and a conclusion or action to be taken called the "then" part. For example, "If the fog ceiling is below 700 feet and the official weather forecast calls for no clearing within the hour, THEN landing is dangerous, will violate air traffic regulations, and diversion to a neighbouring airfield is recommended." To find out if a rule is relevant to the reasoning task at hand, the problem-solving programme must scan over the store of "ifs" in the knowledge base. That search can be immense in the size of knowledge base the Japanese plan to make possible. Here again, the knowledge-base management subsystem will be designed to organise the memory in ways that will reduce the amount of processing to be done.

Parallel processing capabilities in both the software and hardware levels of the system will also speed associative retrievals. In the Fifth Generation plan, knowledge will be stored electronically in a large file known as a relational data base. The job of automatically updating the knowledge in the file, and of organising appropriate searches for relevant knowledge, will be performed by the knowledge-base management software. The Fifth Generation prototype knowledge-base subsystem will handle a modest knowledge base thousands of rules and thousands of objects-about the size needed for current expert system applications. Each object will be allotted 1000 characters of file storage space. Within the ten-year trajectory of their plan, the Japanese goal is to develop knowledge-base capacity in their systems that will be able to handle tens of thousands of interference rules and one hundred million objects! What could so much knowledge encompass?

Since the Fifth Generation is so far-reaching, it demands dramatic improvements in other technologies that support the main-line knowledge information processing systems (KIPS) goals. Essential to the future of the enterprise, for example, are extremely high-speed processors, capable of processing by orders of magnitude faster than anything now available. Artificial intelligence made its debut on first generation machines and has been implemented subsequently on second- and third generation- machines, but not yet on fourth generation supercomputers.

The Japanese aim for chips with 10 million transistors. Chips in current production carry a few hundred thousand transistors at most. Such processors are being developed in the course of another MITI effort, the SuperSpeed Computing Project, and will be adapted into the Fifth Generation machines.

In addition, the Fifth Generation depends on access to knowledge bases in many locations, so its technology will ultimately be fused with the most advanced communications technologies.

The whole area of intelligent interfaces—the ability the machines will have to listen, see, understand, and reply to human users—will require extensive R&D in natural language processing in speech understanding, and in graphics and image understanding. All these have been concerns of artificial intelligence research from virtually its beginning some 25 years ago, and basic research in each of these fields has made reasonable progress. Because non-experts will be the largest group of users, natural language processing is one of the most important research goals of the Fifth Generation.

Research here will cover speech wave analysis, phonetic and syntactic analysis, semantic analysis, and pragmatic analysis, which derives understanding by extracting themes or foci in a given sentence, detecting focus shifts, and so on. For speech output, sentence generation will also be studied. Text analysis is also considered a part of natural language processing by the Japanese, although they are quite aware that the techniques used for large-scale text analysis are different from the techniques needed to smooth the way for an individual user to talk to his machine.

The Japanese see knowledge systems as a potential solution to the steadily increasing amount of text and documents that must be handled by computer. "Our research on intelligent man-machine interface will help to solve this problem," reported a group of Japanese scientists at a Fifth Generation project meeting. Present artificial intelligence research suggests this can be done in a prototype system, intelligent automatic analysis has been successfully applied to a wire news service in the US, for instance-but the sheer scale of the automatic analysis planned by the Japanese dwarfs any existing systems.

All this research in natural language processing will proceed in three stages, beginning with an experimental system, followed by a pilot model implementation stage that is connected with the inference and knowledge base machines, and concluding with prototype implementations. At that point the machines will be expected to understand continuous human speech with a vocabulary of 50,000 words from a few hundred or more speakers with 95 per cent accuracy. The speech understanding system is also expected to be capable of running a voice-activated typewriter and of conducting a dialogue with users by means of synthesised speech in Japanese or English.

The machine's capacity to respond intelligently to users, known as its question-answering system, will first be designed to handle queries in the computing field, but it is expected to be a prototype for such systems in many professional fields: in addition to the query system's 5000 or more words of vocabulary, it will have 10,000 or more inference rules.

PERCEPTION

Picture and image processing are considered almost as important as language processing, especially as they contribute to computer-aided design and manufacture (CAD/CAM) and to the effective analysis of aerial and satellite images, medical images, and the like. Here again the research will take place in three phases, beginning with an experimental phase to tackle ' such topics as the hardware architecture of "feature extractors "for example, to distinguish the boundaries of objects—display generators, and the image data base.

The second phase will produce a pilot model, and in the third and final phase a prototype will be built, integrated into the Fifth Generation machine, and applications will be studied. One obvious use is in constructing robots that can see, understand, and act under novel circumstances. The bulk of robotics R&D, however, will be done in a Robotics National

Project. Eventually, the image understanding system is expected to store about 100,000 images.

There are both positive and negative sides the Japanese project. First, its flaws. The science upon which the audacious Fifth Generation plans are laid lies at the outermost edge and beyond what computer science presently knows. The plan is risky; it contains several "scheduled breakthroughs." There are major scientific and engineering challenges in every aspect, from artificial intelligence through parallel architectures and distributed functions to VLSI design and fabrication. The project demands early successes to maintain its momentum and funding, and that could be a problem.

Conversely, meeting or exceeding the goals of the first three- year periods might well propel the Japanese ahead of their timetable, bringing increased support from the participating companies. Central to the success of the project are the Japanese managers, both governmental and industrial. Generally conservative and risk-averse, they are now to be charged with managing a very ambitious, high-risk project based on technology they hardly understand. Most of the breakthroughs the Fifth Generation project must achieve are basically innovations in software concepts.

The key ideas in the approach to knowledge information processing systems came out of the software world—ideas about the creation, maintenance, and modification of large and complex symbolic data structures in computer memories and the discovery of symbolic lines of reasoning. A quick fix to the problem is to work on the intermediate territory of the so-called firmware—intricate and detailed "programming" of the hardware-switching functions that sit at the bottom of the computing process. This is not a desirable final solution, however, since interpreting and executing the "firmware programme" consumes time and slows down the machine.

Japanese computer specialists and managers are not, and never have been, comfortable with software—it's intangible to them, and its production is notoriously difficult to manage "on schedule and on budget." The Japanese lack the experience base in knowledge engineering and expert systems from which to draw as they begin to work out the details of what to build. In addition, the Japanese lack a large corps of university

Finally, from the Al viewpoint, two elements of the plan are questionable. First, the priority given to the highspeed logic processor. Are all those millions Of Lips needed? Second, is PROLOG the best choice for the machine language of the logic processor? In the American engineering experience, few applications have been limited by the number of inference steps per second that could be performed. Rather, limitations in performance arise from limitations in the quantity and quality of knowledge available to the machine (too little and not well defined); the facility with which it can be managed and updated; and the speed with which it can be searched and accessed.

A second plus for PROLOG is that it solves problems by proving theorems in first-order predicate calculus using computationally fast methods. The user never has to be concerned with the details of the problem-solving process. But PROLOG detractors see this as a serious flaw. The major successes of Al have come from mastering the methods by which knowledge can be used to control the search for solutions in complex problems.

The last thing a knowledge engineer wants to do is abdicate control to an "automatic" theorem-proving process that conducts massive searches without step-by step control exerted by the knowledge base. Such uncontrolled searches can be extremely time-consuming. The parallelism that can be brought to bear is a mere palliative, because the searches become exponentially more time-consuming as the problem complexity increases. One can't keep up with exponential growth simply by lining up a hundred or a thousand more parallel processors.

Creating the knowledge industry, with hardware, software, and knowledge system applications, is a great bet. Indeed, it is one of the few great bets sitting out there now in the information processing industry, ready for a major push towards exploitation. Of course the traditional modes of numerical calculation and data processing will continue to develop and prosper. But these will see steady incremental growth, not explosive growth.

The exponential growth will be seen in symbolic computation and knowledge-based, reasoning by computer. MITI's key economic insight is correct. For an island trading nation, exports create wealth and in the knowledge industry the value of exports is enhanced by indigenous resources-the intelligence, education, and skill of people. Further, knowledge information processing systems will significantly enhance the productivity of many other industries, thereby indirectly contributing to the value added.

The creating of ICOT, the pooling of talent in a cooperative endeavour, plus the well-coordinated transfer of technology between ICOT and the parallel labs of the firms, seem inspired. The ten-year planning horizon is excellent. Ten years is a long time in the information processing industry. Ten years ago pocket calculators cost hundreds of dollars, video games were primitive laboratory toys, and the Japanese had yet to produce their first viable microelectronic chip. As we live through it, we tend to underestimate the speed of technological change. MITI's concern for nurturing the innovative talents of Japanese computer scientists appears well placed. The US and the advanced European nations have become wary of providing the leading technologies upon which Japanese technical achievements have hitherto relied.

Trade wars are under way, and blockades are inevitable. Though solutions to the technological problems posed by the Fifth Generation plan may be hard to achieve, paths to possible solutions abound. The Japanese are rich with excellent engineering talent and have an adequate supply of cutting-edge computer scientists. The mix of talents enables (but does not guarantee) a good chance of success.

THE TRUTH

Managers of the Fifth Generation project have said that it would not disturb them if only 10 per cent of the project goals were achieved; others have remarked that the ten year planning horizon should not be taken too seriously; that the project goals are so important that an extension over another half or full decade would not be unreasonable. Partially realised concepts that are superbly engineered can have great utility and be of great economic benefit. At the very least, a partial success can pre-empt the area and make it not worthwhile for others to enter playing catch up.

The first 20 per cent of the technical achievement may skim off 80 per cent of the potential economic gain. If true, firms in the American industry might never find it in their economic interest to enter the arena. Being late might put them out of the contest. Consider this: though video-taping was invented in the US, the lengthy and expensive R&D process for the consumer-oriented video cassette recorder led to an all-or-none market share result, with American industry getting the "none." No matter how partial its success may be, the Fifth Generation project will provide a decade-long learning experience for a new generation of Japanese computer scientists.

They will be called upon to confront and perhaps solve the most, challenging problems facing the future of information processing, rather than re-engineering traditional systems. They will be learning advanced software concepts in a way that has never been done before in Japan and has never been widely done in the US or Europe. The Fifth Generation project, in its short life, has emplaced the technology transfer mechanisms necessary for Japanese industry to move effectively to bring its developments to market. Right now, the US has a substantial lead in virtually every area of Fifth Generation work.

Of course, the answer depends on what one substitutes for "XXX." Readers who filled in, say, "national socialism" could have supposed that these words were spoken, as a warning to German intellectuals who had not yet appreciated the glory of the Nazi revolution, by Josef Goebbels on the occasion of the book burning in Berlin on May 10, 1933. Readers could substitute "the ideas of the great leader and teacher" for "XXX" and leave open what particular revolution is being talked about. Leaders who come to mind, and whose names would render the quoted paragraph plausible, are, to name just a few: Karl Marx, General Pinochet, Stalin.

The Germans, by the way, had a word for what intellectuals are here being warned to do: Gleickschalffing, which is translated as "bringing into line" or "coordination." But, implausible as it may seem at first glance, "XXX" in the quoted passage stands for "this new instrument," meaning the computer. The authors of The Fifth Generation maintain that intellectuality, the creative use of the mind engaged in study and in reflection, will soon become inevitably and necessarily dependent on the computer. They are astounded that American intellectuals aren't rushing to enlist in their revolution.

A professor of computer science at Stanford University and a co-founder of two commercial companies that market artificial intelligence software systems, and Pamela McCorduck, a science writer, give the reader an idea of what's happening in the world of computers. They make the following claims. First, certain American computer scientists have discovered that, if computers are expected to intervene in some activity in the real world, then it would help, to say the least, if they had some knowledge of the domain of the activity in question. For example, computer systems designed to help make medical diagnoses had better know about diseases and their signs and symptoms.

Second, other American computer scientists have described designs of computers, "computer architectures," that depart radically from the industry's traditional design principles originally laid down by the pioneer computer scientist. The new architectures allow computational chains to be decomposed into steps which can be executed as soon as the data for executing them are ready. They then don't have to wait their turn, so to speak. Indeed, many steps can be executed simultaneously. Computation time is in a sense "folded" in such machines, which are consequently very much faster than their orthodox predecessors.

Third, still other computer scientists, mainly French and British, have created a computer language which they believe to be well suited for representing knowledge in computers in a form that lends itself to powerful logical manipulation. The Japanese hope that the conjunction of this computer language with the new architecture will allow ultra rapid computation of "inferences" from masses of stored knowledge. Fourth, these developments have taken place at a time of continuing dramatic progress in making computers physically smaller, functionally faster, and with increasing storage capacities.

Finally, the Japanese, who already dominate the world market in consumer electronics, have seized on the resulting opportunity and decided to create entirely new and enormously powerful computer systems, the "Fifth Generation," based on the developments described above. These systems, as the book jacket puts it, will be "artificially intelligent machines that can reason, draw conclusions, make judgments, and even understand the written and spoken word." This appears to be a very ambitious claim. But not to seasoned observers of the computer scene, who have long since learned to penetrate the foggy language of the computer enthusiasts.

In other words, the most recent ambitions of the Japanese were already close to being realised according to leaders of the American artificial intelligence community a quarter of a century ago! All that remained to be done-and it would be done within the "visible future"-was to extend the range of the problems such machines would solve to the whole range of the problems to which the human mind has been applied. That ambition remains as absurd today as it was twenty-five years ago. In the meanwhile, however, much progress has been made in getting computers to "understand" the written word and even some words spoken in very highly controlled contexts. Is the Japanese project then really not very ambitious?

A geriatric robot that frees old people from the murderous instincts of their children and is programmed to lie to them systematically, telling them that it understands their petty stories and enjoys "listening" to them. Mechanical doctors we can be utterly candid with and which won't disapprove of us, as human doctors often do. Technical devices in our own homes that gather information about us and determine to what group we ought to belong and which ones we should hate.

Technical systems that permit us to exchange knowledge "engagingly" with our fellow creatures while avoiding the horror of having to look at them or be looked at let alone touched. This is the world Feigenbaum and McCorduck are recommending. In fact, they haven't told us the whole story. Professor Tohru Moto-oka of Tokyo University and titular head of the Japanese Fifth Generation project promises even more:

> ... first, Fifth Generation computers will take the place of man in the area of physical labour, and, through the intellectualisation of these advanced computers, totally new applied fields will be developed, social productivity will be increased, and distortions in values will be eliminated [emphasis added].

We are well on our way to the kind of world sketched here. Already, we are told, and it is undoubtedly true, many people prefer to "interact" with computers. Schoolchildren prefer them to teachers, and many patients to doctors. No one seems to ask what it may be about today's doctors and teachers, or with the situations in which they work, that causes them to come off second-best in competition with computers.

The computer has long been a solution looking for problems the ultimate technological fix which insulates us from having to look at problems. Our schools, for example, tend to produce students with mediocre abilities to read, write, and reason; the main thing we are doing about that is to sit kids down at computer consoles in the classrooms. Perhaps they'll manage to become "computer-literate"-whatever that means even if, in their mother tongue, they remain functionally illiterate. We now have factories so highly computerised that they can operate virtually unmanned.

Devices that shield us from having to come in contact with fellow human beings are rapidly taking over much of our daily lives. Voices synthesised by computers tell us what to do next when we place calls on the telephone. The same voices thank us when we have done what they ask. But what does "the same voices" mean in this context? Individuality, identity, everything that has to do with the uniqueness of persons—or of anything else!-simply disappears.

No wonder the architects of, and apologists for, worlds in which work becomes better and more engaging to the extent that it can be carried out without face-to-face interaction, and in which people prefer machines that listen to them to people, come to the conclusion that intellectuals are irrelevant figures. There is no place in their scheme of things for creative minds, for independent study and reflection, for independent anything.

Aside from whatever advantages are to be gained by living our lives in an electronic isolation ward, and aside also from the loss to "them" of a market "we" now dominate, what good reasons are there for mounting our own Fifth generation project on a scale, as Feigenbaum and McCorduck recommend, of the programme that landed our man on the moon? -if you can think of a good defence application.

One chapter of their book is devoted to "AI and the National Defence." However, the chapter is only six pages long, and is mainly a song of praise-perhaps gratitude is a more apt word-for the Pentagon's 11 enlightened scientific leadership." Laid on thick, the praise tends to betray its own absurdity:

> Since the Pentagon is often perceived as the national villain, especially by intellectuals, it's a pleasure to report that in one enlightened corner of it, human beings were betting taxpayers' money on projects that would have major benefits for the whole human race.

The corner mentioned is the Defence Department's Advance Research Projects Agency, DARPA, also often called just ARPA. Well, there is reason for workers in Al to be grateful to this agency. It has spent in the order of $ 500 million on computer research, and that certainly benefited the AT community (the artificial intelligentsia) enormously. But "major benefits for the whole human race".

The military, however, has good reason to continue, just as generously as ever, to provide funds for work on the fifth generation, as Feigenbaum and McCorduck make clear: The so-called smart weapons of 1992, for all their sophisticated modern electronics, are really just extremely complex wind-tip toys compared to the weapon systems that will be possible in a decade if intelligent information processing systems are applied to the defence problems of the 1990s.

The authors make five points. First, they believe we should look "with awe at the peculiar nature of modern electronic warfare," particularly at the fact that the Israelis recently shot down 79 Syrian aeroplanes with no losses to themselves. "This amazing result was achieved," they write, "largely by intelligent human electronic battle management. In the future, it can and will be done by computer.

Second, we cannot afford to allow the "technology of the intelligent computer systems of the future . . . to slip away to the Japanese or to anyone else.... Japan, as a nation, has a long-standing casual attitude towards secrecy when it comes to technological matters." Third, with the ever-increasing cost of military hardware, "the economic impact of an intelligent armaments system that can strike targets with extreme precision should be apparent . . . fewer weapons used selectively for maximum strike capability."

Fourth, "it is essential that the newest technological developments be made available to the Defence Department." Finally, "the Defence Department needs the ability to shape technology to conform to its needs in military systems. Here, perhaps more than in any other argument of the book, we are close to what it's "all about." A much shorter, and better, account of the Japanese Fifth Generation project appeared recently as a cover story of *Newsweek magazine*.. People generally should know the end use of their labour.

Students coming to study at the artificial intelligence laboratories of MIT, my university, or Stanford, Edward Feigenbaum's, or the other such laboratories in the US should decide what they want to do with their talents without being befuddled by euphemisms. They should be clear that, upon graduation, most of the companies they will work for, and especially those that will recruit them more energetically, are the most deeply engaged in feverish activity to find still faster, more reliable ways to kill ever more people-Feigenbaum and McCorduck speak of the objective of creating smart weapons systems with "zero probability of error" (their emphasis). Whatever euphemisms are used to describe students' Al laboratory projects, the probability is overwhelming that the end of their research will serve this or similar military objectives.

To underline the importance of the computer as the "main artifact of the age of information," Feigenbaum and McCorduck instruct us just wherein the importance of the computer lies: [The computer's] purpose is certainly to process information-to transform, amplify, distribute, and otherwise modify it. But more important, the computer produces information. The essence of the computer revolution is that the burden of producing the future knowledge of the world will be transferred from human heads to machine artifacts [emphasis in the original].

How are we to understand the assertion that the computer "produces information"? In the same way, presumably, as the statement that a coal-fired electric power station produces energy. But that would be a simple and naive falsehood. Coal-fired power stations, transform energy, they do not produce it. Computers similarly transform information, generally using information-losing operations. For example, when a computer executes an instruction to add 2 and 5, it computes.

EFFECT ON SOCIETY

Since then, it has had some notable successes, enabling computers to perform-albeit in a very limited way some of the tasks normally done by our minds. Some Al workers see Al as a way of helping us understand human psychology; they try to write programmes that tackle their tasks in the sort of way in which we do. Others see it as an approach to a theory of intelligence in general, human (and animal) intelligence being a special case. Still others simply want to write programmes to do something (to understand language, to describe visible objects, or to solve problems, of various kinds), irrespective of how we do it.

And most of these hope that what their programmes do will be not only interesting, but useful. The technological aspects of Al have suddenly become more visible. Public interest in Al, and media coverage of it, have increased enormously over the last two years. More and more people view it as an incipient technology of great potential power and social significance. The public interest dates from the announcement in 1981 of Japan's ten-year national plan for developing "Fifth Generation" computers. These are defined as incorporating AI techniques (as well as large-scale parallel processing).

Since then, large sums of money for AI-research have been made available also by governments and industry in the Western industrialised nations.