Computer Science
University of Manchester

The Global-yet-Personal Information System

J. R. Gurd and C. B. Jones

Technical Report UMCS–94–4–1 (revised August 1994)

The Global-yet-Personal Information System

J. R. Gurd and C. B. Jones*

Department of Computer Science
University of Manchester
Oxford Rd., Manchester, U.K.
jrg@cs.man.ac.uk, cbj@cs.man.ac.uk

August 19, 1994

Abstract

The editors of this volume[1] set out to compile a set of personal views about the long-term direction of computer science research. In responding to this goal, we have chosen to identify what we perceive as a long-term challenge to the capabilities of computing technology in serving the broader needs of people and society, and to discuss how this 'Grand Challenge' might be met by future research. We also present our personal view of the required research methodology.

Technical Reports issued by the Department of Computer Science, Manchester University, are available by anonymous ftp from `ftp.cs.man.ac.uk` in the directory `/pub/TR`. The files are stored as PostScript, in compressed form, with the report number as filename. Alternatively, reports are available by post from The Computer Library, Department of Computer Science, The University, Oxford Road, Manchester M13 9PL, U.K.

[1]This report is a draft of a contribution to *Computing Tomorrow*, Ian Wand and Robin Milner (editors), Cambridge University Press. As a courtesy to the publishers, all citations should be to that source.

Contents

1 Introduction

Much of present-day computer technology is concerned with the processing, storage and communication of digital *data*. The view taken in this contribution is that a far more important use of computers and computing is to manage and manipulate human-related *information*. Currently, the provision of such structured information has been tackled at the level of single organisations (company, institution, government department, etc.) by the use of *databases* which are often limited to single functions within the organisation. Databases are *closed*, in the sense that the information itself can be viewed in a limited number of ways, and the ways in which it can evolve are carefully controlled. Interaction between databases containing related data is prohibited, except through the mediation of human experts. This is an unnecessarily restricted concept of information processing, and one which fails to recognise its real social and economic potential. We foresee a huge market for *personal* information services based on *open* access to a continually evolving *global* network of stored information. Although there are significant technical difficulties associated with creating and controlling such networks and services, we predict that the economic incentives will ensure that the necessary development occurs and that this information market will – within a period of decades – dwarf the market in computing machinery and software.

With regard to the research methodology, we speculate that computer science is evolving into the more general discipline of *informatics* and consider the nature of informatics as an emerging science.

1.1 The Importance of Information

Information is an extremely potent commodity whether for individual humans or for organisations. Time and again we have seen cases where access to the right information at the right time saves lives, creates wealth, improves the environment, etc., etc. The contribution made by Gutenburg to the dissemination of information is discussed in the seminal work of Marshall McLuhan [McL62]. The exponential progress in science, technology and engineering would have been impossible without our ability to disseminate technical information. Today's computing and communications technology can support an enormous growth in information access, inducing who knows what changes in our society. However, two things threaten further progress: one is the worrying question of whether we will actually drown in the amount of undigested data which is now available to us; the other is whether, in spite of the potential availability of information, we simply cannot gain access to it, either because we are unaware of its existence or because there is no structured way to search for the items in which we are interested.

CD-ROMs containing enormous quantities of data are advertised in the national press; the Internet is an international network whose capacity is growing at a phenomenal twenty percent per month; an enormous amount of raw data is available as strings of ASCII characters over the Internet; use of the 'World Wide Web' is growing at a rate which beggars the imagination. It is tempting to imagine that we live in an information-rich age, since there is this potentially enormous wealth of information that can now be accessed electronically. On the other hand, unnecessary difficulties impede legitimate use of electronically controlled information to such an extent that individuals could consider themselves at the same time information-poor.

In order to understand this paradox, it is crucial to distinguish between information and the raw data. In this article, we use the two terms as follows:

data — uninterpreted byte streams

Raw data is stored on magnetic media. It is a slight simplification to say that it has no structure; in fact, the byte stream can contain bytes marking some structure, but this is only given meaning by execution of specific programs. An interesting example of this is the markup notation provided in SGML (Standardised Generalised Markup Language).

information — collections of named data items, linked – for example – by relations

Information about information structures is accessible by programs: in many database systems, the data dictionary is itself accessible.[2]

Such information – in the above sense – that we do have access to derives from *databases* stored in digital computer systems. Such databases act as passive 'data warehouses', containing a vast quantity of raw data that has to be carefully garnered and then painstakingly interpreted by human 'users' before it can be put to good use. Programs which actively seek information can be envisaged and, indeed, some exist, but they are currently primitive. Enquiries which require access to more than one database face almost insuperable problems of *incompatibility*, in interface protocols, data organisation, data formats, etc.

In order to amplify this criticism, let us first exemplify the positive ways in which computer systems have already managed to transform our lives. Rather than a factory filled with robots, consider the impact of 'EPoS' (Electronic Point-of-Sale) on a supermarket belonging to a large chain: it no longer needs to carry most of its stock in its storeroom; computer tracking of sales, plus planning of deliveries, enables the shelves to be restocked more-or-less directly from lorries. This ensures fresher produce and less wastage, and reduces the cost of holding stock. Stocking of the warehouse and dispatch of vehicles are also controlled by computer. When customers go into the supermarket, they collect goods whose identity is recognised via a barcode ('EAN', in the UK) which is used both to locate the price for the customer's itemised bill and to update the stock level; the customer has the option of paying for the purchases via a transaction, controlled by a plastic card, which ultimately ends up in a change in bank balances (one of many transactions carried out without the intervention of a human being).

1.2 Problems in Accessing Information

Limitations in the benefit of computer systems result from the difficulty of accessing and/or updating information, and this is equally easy to exemplify. There are an enormous number of examples where information which *is potentially available* somewhere in the world is either difficult or impossible to access. Even though we would be prepared to pay for access to such information, and it would increase the effectiveness and/or fulfillment of our lives, we find ourselves unable to get it; indeed, all too often it is only the notorious 'hackers' who are prepared to expend the necessary time and ingenuity to access useful information.

An example which again affects most individuals is the need to plan journeys. Let us look at what is currently available. Some information – such as flight delays – is freely available on 'Teletext' or 'Oracle'. It is possible to go into a travel agent and to discuss with a human agent the plans for, say, an airline trip. Travel agents normally use computers which enable them both to see the potential routings and the status of bookings on various flights. But there is in fact no reason why we need to go to the travel agent. Many of us have sufficient access to information processing facilities that, if we could only access the database directly (as

[2] For the sake of brevity, we shall frequently use the term 'information' to mean, strictly speaking, 'information embodied in [the] raw data'.

the French can for train bookings via 'Minitel'), we could handle the booking ourselves. More importantly, many journeys are not straightforward cases of travelling by air. What we would really like to do is to look at a range of possibilities comparing (combinations of) train, car and air routes: it would be desirable to find out about available trains without having to log on to a totally separate database; one would also wish to know about the economics of renting a car and undertaking part of our journey by road (and the probable conditions of the roads); at the same time, we would like to be able to access information about the availability of hotel rooms at prices acceptable to us, and perhaps their vicinity to starred restaurants! If all of the data were in one database, this programming task would be well within the state-of-the-art. It would not be difficult to write a computer program which also stored information about our preferences and which, on being given the starting points and finishing points and approximate times of our desired journey, would come back to us with a range of options ordered by our stated desires. But the crucial difficulty is that up-to-date information can only be obtained by accessing *a range of* relevant databases.

Of course, end-users do not always act solely in a personal capacity. There are innumerable examples where corporate users of information systems would like to have access to a range of related information at the same time. For example, a company which is planning new water treatment plants needs an enormous range of information: firstly, demographic information, especially projected changes in population; it needs information about the terrain in which potential water storage facilities could be built; weather information needs to be correlated with geographic information about rivers, including their rates of flow; perhaps information is required on the state of pollution in those rivers, in order to plan further processing facilities; and so on. Much of this information *is available* in databases and, given network access, could be obtained in seconds. The challenge faced by anybody wishing to undertake such an engineering project is to access information from a variety of sources in a range of information structures, to correlate it and present it appropriately, probably to a fixed and tight time schedule, in order to facilitate rapid planning. There is in fact a huge economic advantage to an engineering concern in being able to access and process information quickly.

One can find examples where there are even larger financial consequences, and where immediate access to a wide variety of accurate information sources can give the decision-makers a huge advantage over their rivals. For example, multi-national companies about to take long-term investment decisions might benefit from access to information about investment support and taxation incentives, etc. around the world; or governments may wish to base decisions on interest rate changes on the instantaneous state of the economies of other countries.

Everyone can benefit from access to multiple sources of computer-based data, suitably interpreted as the human-related information it represents. Such information would be even more useful if it could be integrated in a reasonable way [MGT+87]. However, there are severe technical problems standing in the way of this ideal.

Our contribution to this volume is founded on the twin beliefs:

1. that making *all* such stored information potentially and readily accessible, in integrated form and on an individual basis, will create a potent new driving force for the global economy; and

2. that reaching such a position constitutes a 'Grand Challenge' for research in computer science (or, more generally, informatics).

1.3 Background

As we have pointed out already, the state-of-the-art is represented by database technology, which is closer to data management than information management. Furthermore, most of the effort in developing databases so far has been devoted to compiling the data. Even this has been done only on an *ad hoc* basis; the data is gathered by whatever means are available, and then humans are challenged to gain access to it. Increased processing power can be thrown at the human-computer interface to overcome the inconveniences resulting from the *ad hoc* gathering of data but even this is insufficient and a new breed of employee, the 'database access adviser', has had to emerge to help the real 'end-users' gain access to the information they really want. The scientific activities of chemists are now supported by hundreds of databases; in many cases, access requires support from advisers. In office management terms, while computers may have supplanted filing clerks, they have not been able to replace administrators responsible for search 'strategy'.

It might be argued that this approach at least has the merit of simplicity, thereby minimising the opportunities for human error. However, even in the simplest systems of today, numerous anecdotes serve to warn us that any strong faith in our information systems is misplaced. The growth of electronic news boards and news groups is itself a fascinating example of information dissemination routes which were unthinkable twenty years ago. One such news group (`comp.risks`) records some of the horror stories which occur when computer systems are poorly designed or used in inappropriate ways. A common cause of many of the unfortunate experiences seems to be our inability to distinguish the information in our databases from the situation in the 'real world' that it purports to represent.

Little can be done to protect a system from being given *incorrect* information (that is, information that incorrectly represents the external reality). For example, people whose death is inaccurately reported can no longer resolve the mistake as readily as Mark Twain's humorous cable to the Associated Press; they can find that they are no longer allowed to hold a driving licence and they become a non-person as far as the relevant computer system is concerned; they can even be prosecuted for driving in this state (although it is presumably possible that they might avoid having to pay tax)!

The problems of keeping the abstractions inside a computer in step with the reality of the world about which they are supposed to record information are enormous and often revolve around issues of uniquely identifying objects which are reviewed in [Ken78]. While writing this article there was a newspaper report that the UK Driver Vehicle Licencing Agency had been investigated by the National Audit Office and it was found that *one third* of the records about drivers and *one quarter* of the information about vehicles contained errors.

A more subtle difficulty is that stored information may be *unverifiable*. For example, when bank customers call into question so-called 'phantom withdrawals' from their accounts (via ATM bank terminals), banks often deny that this can happen and the very nature of electronic withdrawal makes resolution of the dispute impossible.

Even if we can trust the stored information, it may only be available in a form that is *incompatible* with our needs. For most databases, only certain stylised interactions retrieve information in exactly the right form for immediate and 'natural' use. Such interactions usually correspond to the circumstances specified in the original system design (although many systems do not even satisfy their design criteria). When circumstances change, and information has to be accessed in ways not foreseen in the specification, new problems can arise. For example, the retriever may make unwarranted assumptions about what the information means.

A ubiquitous problem is that of properly identifying or naming an intended entity. To cite but one example: one of the authors, on publishing his first UK book (*Software Development a Rigorous Approach*), received a letter from the British Library requesting information (e.g. date of birth) and asking whether he was the same Clifford Jones as had written *A Companion to the Sunday Service*! This kind of mistaken identity is easy to understand, but difficult to do anything about. The consequences might be less amusing if the database involved information about, for example, terrorists.

Another important consideration is that certain kinds of trustworthy information may be *inaccessible*, either because access is forbidden (e.g. by law), or because access is impossible by design, or because access simply costs too much (in terms of either money or time).

Note that these difficulties are associated with relatively simple and self-contained information systems: the warning is that even these can misfunction.

1.4 Summary

It must be clear by now that we see the crucial challenges for the future of computing to be in the compiling and accessing of human-related, symbolic information. This is in contrast to the view of computing, typified by the US president's High Performance Computing and Communication initiative, which believes that important advances in the past have been due to the provision of ever more computational power, and that market growth depends on more of the same. Although many (physical) scientists and research engineers seem able to absorb – in a Parkinson-like way – any amount of computing power which is made available to them, they represent a particularly narrow aspect of the economy and society at large.[3] We believe that the real potential of computers both for the average member of the general public and for institutional decision makers has not yet been, and will not be anything like, realised until facilities are available for 'mining' large bodies of information which are widely dispersed. This also contrasts with the way in which databases are viewed at present; they are usually designed for a single, organisation-oriented purpose, and to be 'closed' to the outside world.

Hence, wherever information may be stored, we wish to have access to it and the ability to use it for our own purposes. This implies that we want to view our information system as being at the same time

global — information stored in the totality of all physically distributed repositories should be potentially accessible, subject to provision of the physical means to access the raw data;

open — except where there are proprietary reasons for secrecy, access to all information should be both allowed and encouraged – subject to payment at the appropriate market rate, anybody should be entitled to access the information for whatever purpose they desire; and

personal — information should be structured in a way which facilitates access and individuals should be able to obtain access in a way which fits their stated preferences.

The system meeting these requirements would be known as the Global-yet-Personal Information System: **GyP∫IS**.

[3] But we must not lose sight of the way in which embedded computing has grown: television sets are now being designed with 0.5 Mbyte of control software, hand-held telephones with 1 Mbyte and even the humble shaver appears to need 2 kbytes of software!

Because of our change of focus from computation to information access we prefer to use the term *informatics* rather than computer science. We understand by this term study of the capture, storage and retrieval of information as well as the more traditional, computational aspects of our subject.

2 The Global-yet-Personal Information System

$\boxed{\text{GyP}\!\int\!\text{IS}}$

This section explores the technical, legal, social and economic issues governing the provision of global, open and personal information. We commence by defining some terms.

2.1 Terminology

A traditional view of a computer system is that it entails the following five activities:

- input (capture of data from human sources);
- storage (of data);
- processing (changing or rearranging data);
- communication (physical movement of data); and
- output (delivery of data to human 'users').

However, from the *user's* point of view, the *interaction*, via input and output, is the only thing that matters. The fact that storage, processing and communication are needed is a property of the internal system, and ideally should be invisible to the user. Yet, in practice, these matters are only too visible to the user. For the moment, we shall concentrate on the user's view; we shall focus on information rather than data, and pretend that decisions on storage, processing and communication are hidden. To mark this change of view, we will use the terms *compiling* and *accessing* of information, in place of input and output of data, respectively. We shall use the term *information system* rather than computer system, and we shall refer to an information *repository* rather than a database.

Two examples should suffice to show how this view can define human-computer interactions in diverse application areas.

Query Processing — In a relational database, the processes of updating internal information and making queries are well defined, and are the only actions entirely visible to end-users. Of course, for queries to be *efficient*, someone needs to know how the data dictionary is structured, but this is usually a computer specialist, rather than a 'true' end-user: we term such a specialist a *moderator*.

Numerical Simulation — It is traditional to think that other application areas, such as scientific simulation, are fundamentally different from query processing, but there are in fact many similarities. Consider, for example, numerical weather prediction: information about the current global state of the weather is gathered by a mixture of human and electronic means, and this provides the initial state for a 'time-stepping' simulation that iterates until it has computed the projected state of the weather at some required time in the future. The projected state is then post-processed into a form suitable for human assimilation (e.g. via graphical 'visualisation'). The details of the simulation (and the gathering of starting data plus the interpolation of the initial state) are of concern only to physical scientists and computing experts: the immediate end-users, i.e. the weather forecasters, wish to concern themselves only with interpretation of the simulated results of the physical model. Of course, the 'true' end-users are the people who rely on the outcome of the forecasts. Even though this example involves a considerable amount of processing (in the simulation), it is still, from the user's point of view, a matter of first compiling the necessary input information and then accessing the formatted results.

Another motivation is provided by Vannevar Bush's [Bus67] picture of the scientist's desk: 'Memex', which clearly recognises the importance of connections between items of information. Using this as a model, we can begin to imagine what a 'virtual classroom' might be like as a networked learning environment: we expect such an environment to be more important than libraries to future generations of students.

2.2 Technical Difficulties

We have already mentioned some of the difficulties inherent in trying to manage and manipulate information in traditional, closed, database systems: these problems are exacerbated in the GyP ∫IS. Although the Internet is providing access to a phenomenal collection of ASCII files (often structured by HTML), there is only limited provision for access to structured information. Some awareness of the difficulty facing one of the oldest publishing ventures is indicated by the conference being organised in 1994 on *The Future of the Dictionary*.

The global system requires access to information stored in physically distributed repositories. The problem is not primarily that of compiling the information, since so much is already available. So the technical challenge for informatics lies in developing techniques that allow uniform and effective access to the immense variety of existing and yet-to-emerge repositories.

At the most trivial level, there is the problem of data formats. The authors of this article chose to work mainly on Unix systems which are linked together by a departmental Ethernet but, at some stages of the development, they wished to edit some of the text on Mac Powerbook machines. One way of transferring data between the two file systems was to write floppy discs in a third (MS-DOS) format! Although software exists to 'smooth' such transactions, it does not hide completely the multiplicity of underlying data formats. Indeed, it can sometimes cause even more confusion by failing in an obscure fashion as it tries unsuccessfully to hide one thing too many.

There are major technical problems associated with identifying in a global network exactly which piece of information is being referred to by a query. It is well known that it is inadequate to refer to people in a large organisation solely by their names; extra qualifying information, such as date of birth and/or place of birth, is used in order to try to disambiguate references. However, in the realm of the sort of automatic computer system we are envisaging, it is almost certain that programs will have to be capable of handling dialogues in order to ensure that the required entity has been located. An example here is that, whereas the name 'Cliff Jones' is currently unique within the department in which the authors work, it is far from unique in particular geographical areas within the UK.

Moreover, because the information contained in the global network is continually evolving in a distributed fashion, there can be no concept of an instantaneous 'snapshot' of the state of the information. Hence, it is impossible to conduct any 'exhaustive' searches to determine whether a particular entity exists at a particular time. To take another example, the name 'John Gurd' could well be unique in the world at this moment, but there is no tractable way of establishing this as a fact. One of the main reasons for the present dominance of 'closed' information systems is that they can be (and are) designed so as to avoid these problems.

Another problem with multiple, distributed repositories is that they rarely yield the required information in a 'ready-to-use', uniform fashion (note that this is a different problem from the incompatibility of raw data formats). Even if correct information is stored, correlation and integration between different sources often proves impossible and, even where it is feasible, it requires the concerted assistance of a variety of highly skilled information access advisers (note that these have different skills from database access advisers, although their job descriptions are similar). Some of these are what we might call 'repository-oriented', in that they are closely associated with the repository and are knowledgeable about how to make queries about its information efficiently: these are akin to the *moderators* (and database access advisers), introduced in Section 2.1. Other advisers are predominantly 'user-oriented', in that they understand how the retrieved information is to be used, and hence how it should be presented, regardless of its raw format: we call these *presenters*. Yet other advisers fall in-between these two extremes, and are capable of viewing information from both the user and the repository end: these are normally employed by the end-users to search out the required information, so we call them *seekers*. If all these advisers are to be replaced by some form of computation, significant technical developments will be necessary since performance comparable to humans would require far greater *adaptability* of 'programmed' behaviour than has been exhibited hitherto in information systems.

Moreover, it is crucial for the kind of applications we have identified that information is accessed and correlated while it is still 'current'. The moderators, seekers and presenters will have to work to hard 'deadlines', leading to further technical difficulties associated with deciding how much time and 'effort' to devote to searching for, and then processing, relevant data and information. Reliable notions of 'expedience' and 'approximation' will be needed.

A final interesting problem is the effect of the 'longevity' of information on attempts to integrate multiple information sources. Many companies have gone through generations of computer systems which have provided progressively more performance and better economics. They frequently face the problem of needing to access 'old' information which they wish to correlate with 'new' information, perhaps for legal reasons. In the aircraft industry, for example, it is necessary to be able to determine the source of materials in aircraft which were built decades ago; decades in computing represent several generations of hardware and quite probably several generations of database technology. The problems which, for example, insurance companies face in accessing old records are equally severe.

In this context, it is interesting to go back into the history of information systems *within* individual corporations. When computers were first commercially available in the 1960s, many large organisations, such as the major motor car manufacturers, designed computer systems to support different parts of their operations. A computer system might be introduced by a company to control order processing; then a different computer system would be designed to control mechanical production; a third system might be used to control painting, and another system for controlling the stocking of cars; yet one more system could be needed for the forwarding of cars to the manufacturer's dealers; etc. As each need was identified, so a new system was developed to 'satisfice' [Sim69] it. It was not long before senior management realised that this multiplicity of different computer systems was not only inefficient but that it was leading to new sorts of 'clerical' problems which had not been encountered in the manual information processing age. The availability of more powerful and cost-effective hardware, together with the advent of database software, started a trend towards greater integration of information systems. However, it was by no means easy to integrate systems that had been developed separately. It quickly became apparent that even such basic concepts as 'when a car has been sold' were interpreted differently by users (and hence the designers) of the separate information systems. Whereas human beings were able to react to such inconsistent views of the company's operation, their electronic counterparts could not. In fact, more than one corporation failed in its first attempts to develop integrated information systems.

Today, one still finds [AMR94] (see also [PB94] in the same journal) that:

> The U.S. Department of Defense (DoD) currently maintains more than 1.4 billion lines of code associated with thousands of heterogeneous, noncombat information systems ...the department is often unable to obtain correct information from the data stored in the various existing databases due to a lack of standardised data and data structures across systems. Submitting the same query to each of the more than 20 payroll systems can result in not just multiple answers, *but in multiple kinds of answers.* At times, consolidating the query responses has proved to be an impossible task.

All-in-all, we must recognise that the creation of suitable means for flexible access to global information poses significant scientific and engineering research challenges for the emerging discipline of informatics.

2.3 Economic, Social and Legal Issues

The challenge extends outside informatics, to society at large and, in particular, to economics and the law. For example, a primary vision of the GyP∫IS is that it will form a 'driving force' for the economy. People and organisations will be prepared to pay well for information services. This kind of reasoning has been applied previously, for example, in the case of the French Minitel system for the provision of telephone-based information services. In fact, the return on sales of Minitel services has been such that the initial decision to *give away* all the necessary hardware infrastructure has been entirely vindicated. Even more than in the recent past, manufacturers will have to consider the value of the services provided, rather than the material products produced. Just as software creation now absorbs far more of a developed country's GNP than hardware purchase, we predict that within a few decades the new open information 'market' will dwarf both of these.

Society will need to consider the wider implications of an open information market. Once proprietary motivations for keeping information secret have been overcome (and we believe that the economic advantages of exploiting information may well cause this to happen), detailed consideration of the whole concept of ownership and regulation will need to be undertaken; issues of responsibility and liability will have to be resolved (and one can only hope that this is done with more understanding than was evident in the UK Data Protection Act). The dangers of 'Wire Pirates' are discussed in [Wal94].

On the social front, the displacement of human advisers by computational activities, plus the emergence of jobs in the new information service 'industry', will precipitate yet another round of 'reskilling' of the workforce. The effects of this need not be negative: for example, the

French Government has estimated that more than 300,000 new jobs were created as a result of the introduction of the Minitel system.

There are also technical challenges which follow from obvious legal constraints. For example, it is easy to make a case that no information should ever be destroyed (simply new versions created), and this poses further complications for referring to specific items unambiguously.

As an amusing circularity, some of the most commercially exploited, machine-readable, information sources are already those used by lawyers. But, whereas these are paid for by subscription, the issue of how one will pay for (relatively diverse) sources of information is itself a problem which will have to be solved by the 'information industry'.

2.4 Summary

Briefly, then, we envisage the development of a global and open information system that responds quickly and that is tailored to our personal requirements. In parallel, we anticipate the evolution of society to a state in which such a system will 'fit'. Each aspect poses substantial challenges: evolving a strategy to overcome these is what we call the **GyP∫IS Challenge**.

3 The States of the Arts

The previous section has set the overall objective; we now consider this from several different technical perspectives. The choice of these particular viewpoints is based primarily on our perception of the structure of present-day computer science (and is reflected in the range of topics covered by the other contributions to this volume). However, they also suggest to us a rationale behind the emerging discipline of informatics that we feel motivated to investigate further in Section 5. The comments in this section are intended to identify research challenges rather than propose solutions.

3.1 Programming View

A prevalent view in computer science is that software can be developed to solve any problem: programming languages are Turing-complete, so programs can be written to solve any computable problem. Hence it might be thought that the GyP∫IS Challenge could be met by developing appropriate programs.

To see that this view is naïve, one can look back to the evolution of the interface between programs and data. Initially commercial systems were constructed which processed sequential files; these files were commonly on magnetic tapes. The interface between the program and the files was by *read* and *write* statements which brought individual *records* into the store of the computer, processed them and then moved them out again. With the advent of large direct access stores (e.g. discs), database systems began to evolve where programs could obtain direct access to records via *keys*. This led to a generation of online systems, but computer programs were still written to read, process and replace records one-at-a-time. It is only gradually that programs and database descriptions have come to be viewed as being expressible in the same language. There are now machines which have an addressing space which goes across the whole of the memory, including the filestore (so-called 'one level stores'). These make it much easier for a program to be viewed as manipulating data whose description is stored within the program. An interesting comparison can be made with so-called 'persistent' programming languages, such as Smalltalk, where a naming structure is used which makes all data of interest directly addressable by the program.

The GyP∫IS Challenge adds a new dimension to this problem in that it is necessary to access information from many different sources. The discussion of 'Open Systems' in [HdJ84] and the recently proposed concept of 'Megaprogramming' [WWC92] seem to begin to address this need. In the latter approach, integration of accesses to multiple information sources is viewed as a mega-programming task: in order to facilitate expression, the mega-programmer is provided with a richer-than-usual set of interface routines, and a careful procedure is described for installing new or modified mega-modules. The problem with this approach is that only the mega-programmer can respond to change. This contravenes our requirement for adaptability *within* the electronic part of the information system. Nor do we believe that a mega-programmer would be able to anticipate changes at design-time and program the response to these. It is always the unforeseen changes in the environment that cause the greatest problems.

3.2 Database View

Of all the constituent parts of traditional computer science, database research ought to be the most germane to the GyP∫IS Challenge.

One popular database design approach takes the (not unreasonable) position that information consists of *entities* and *relations*. These entities and relations are simply abstractions of the things which surround us in the physical world, so we can think of entities as people, places, geographic data, flight data, and we can think of relations as facts which link entities together, such as that a certain person is booked on a particular flight, that a certain flight is expected to arrive at a certain time, etc. Indeed, this simple entity-relation view is one way of designing closed information systems (see, for example, [RBP+91] or [CAB+94] for a description of its use in 'OMT' or 'Fusion', respectively). Relational databases cement this view as a way of storing and accessing information. In spite of its success, as measured by the use of relational database systems, it should be regarded as no more than one example of a way of structuring information.

Even within closed information systems, there is a challenge of ensuring that the expected connection is preserved between the 'artificial' entities and relations inside the information systems and the 'real' entities and relations in the world about us. This is exactly what goes wrong when it is falsely recorded that somebody has died: the relation in the information system no longer corresponds to the perceived reality. Michael Jackson [Jac83] proposes an approach to designing information systems where the physical world is first understood and then appropriate methods are designed to link changes to its abstract representation in our computer system to events in the physical world.

It is important to realise that the design of such a system changes the world in which we live: future generations of system have to reflect the fact that the transactions in one system have become part of the physical environment for future information systems. As has been observed by Manny Lehman [LB85], the presence of an information system changes the environment and effectively guarantees that the users' requirements will change. For example, consider the way the world has changed now that we can use our 'Switch' (Electronic Funds Transfer) card in the supermarket. In this context, even the old abstraction of bank notes has disappeared: the customer's bank balance is reduced, the supermarket's bank balance is increased, and no 'paper' ever moves between the relevant banking institutions.

Where current database technology lets us down is in coping with open information access.

The notions of a pre-defined data dictionary and fixed ways of locating entities are so ingrained and central to accessing the contained information that they dominate all description. Any access that fails to conform with this in-built structure is destined to fail (usually in an obscure fashion). It is revealing that the 'Human Genome' project has had to make a major investment in development of novel database technology[4].

The companion chapter in this volume by Peter Gray, on 'Large Scientific Databases and Knowledge Reuse', illuminates the way in which 'the database view' may evolve towards the GyP∫IS goal. In particular, it foresees wider reuse of knowledge achieved via object-oriented database technology and further development of 'mediator' and 'facilitator' objects that fulfill some of the functions of our seekers, moderators and presenters.

However, the development of 'better' database technologies is not of itself going to solve all the GyP∫IS problems. As articulated so clearly by William Kent [Ken78], it is always going to be necessary to view data from some unanticipated angle in order to maintain compatibility with the human view of the information it represents: there is no all-pervading, 'best' view. More recently, Brian Gladman has made the insightful observation (in the form of a comment to one of the authors) that 'information is the connection between raw data and purpose', thus making explicit the reason why we should expect the most appropriate view to vary from time to time.

3.3 Artificial Intelligence View

Artificial Intelligence (AI) aims to throw light on human cognition by developing programs that exhibit ever more realistically human behaviour. The 'strong AI' view anticipates eventual development of a simulated human which, given the way some humans are already able to solve many of the problems mentioned above, would appear to address our desire to develop an open information system.

For a while, AI research relied on extensions to mathematical logic and database technology, but it has now incorporated neural models and other forms of novel machinery to try (unsuccessfully, so far) to overcome the limits on adaptable behaviour. While not achieving anything remotely like overall human behaviour, AI has developed programs that can mimic important components of the human system; for example, speech generation and (to a limited extent) recognition. Some databases nowadays can be accessed via a 'natural language' interface, and there have been several attempts to provide adaptive database interfaces (e.g. [HK91]), although the degree of adaptability is very limited when compared with human behaviour.

Attempts to mimic the adaptability of 'real' human behaviour have foundered so far on issues such as *consciousness* and *common sense*. Programs have been developed that can 'plan' activities in relatively restricted scenarios, but learning and responding to change is extremely rudimentary. The state of the art in this area is captured pertinently in the July 1994 issue of the *Communications of the ACM*, on 'Intelligent Agents'. A paper in that issue on the Cyc system [GL94] contemplates potential applications that are highly relevant to our quest, and the Editor's interview with Marvin Minsky [Rie94] throws useful light on the practical barriers to be overcome in future research. This interview also echoes our view that progress in this kind of endeavour relies on integrated application of multiple technical approaches.

Distributed AI and Cooperative Distributed Problem Solving [DLC89] are developments of this topic that seem pertinent to the GyP∫IS Challenge. Generalised languages and systems are

[4] In fact, the entire Human Genome Initiative is redolent with information-oriented challenges that have a considerable amount in common with the themes of this paper [Fre91].

being developed for support of 'multi-agent' computing, and there is interest in the creation of 'societies' of such computational entities (see, also, the comments on [Hub88] in section 3.4).

Of course, it is tempting to believe that achievement of 'strong AI' would offer a way of overcoming all of the problems in the GyP∫IS Challenge. It is certainly true that human beings are much more adaptable than any current form of computation when they have a source of information available to them and are asked to process it in new ways or provide new sorts of reports. Hence, the argument goes, if one can mimic human intelligence, one will solve all such challenges. However, we remain unconvinced that artificial intelligence on its own will offer a complete solution, and we note that the companion chapter in this volume by Alan Bundy, on 'Prospects for Artificial Intelligence', takes a similar view.

3.4 Computer Architecture View

There is another computer science viewpoint that considers all difficult problems to be solvable by provision of sufficiently powerful computer hardware. This has led to the current attention being paid to massively parallel computer architecture. Is there any evidence that such research can meet the GyP∫IS Challenge?

In fact, the fundamentals of computer architecture are challenged by its requirement for a global view of information. Individual repositories in the GyP∫IS will be huge: furthermore, they will be widely distributed geographically. Many of the problems encountered when handling distributed computational objects have been studied in the field of distributed computing. The view is taken that computational *processes* can invoke other computational processes in physically distant processors. Naming conventions are necessary to locate remote processes and processors, and interface protocols have to be adhered to: as mentioned earlier, there is a major problem with identifying exactly which entity is being requested.

In terms of the system architecture, there are important questions about the way in which concurrent and distributed computations are created and managed, and how the necessary communication and synchronisation between them are achieved. The companion chapter in this volume by Roger Needham, on 'Computers and Communications', ponders on the way some of this technology might develop and also speculates amusingly on possible applications of the developing technology over the relatively short term.

The present state of the art in this field is exemplified by the Internet and the World Wide Web. These implement a truly global network of interacting computers, but the degree of sophistication implied by the GyP∫IS is far from being achieved. Nonetheless, there are clear signs that researchers (both implementers and users) are beginning to view this system as an information resource, as exemplified by some of the articles in the August 1994 issue of the *Communications of the ACM* on 'Internet Technology'. In particular, the paper on 'Resource Discovery' [BDMS94] describes situations similar to those expected in the GyP∫IS.

Access paths must be planned, with a view to both performing computation where it can be undertaken most efficiently and to avoiding the transfer of huge volumes of data over expensive networks where most of the data will simply be rejected in subsequent processing, prior to delivery to the end-user. This has obvious resonance with the design of cache memory systems for massively parallel computer systems (see [Tok90] for an interesting view on this). Huberman's book [Hub88] 'The Ecology of Computation' has chapters on 'Agoric Systems', 'Deals among Agents' and 'Market-like Task Schedulers' which all relate to this topic.[5]

[5] As a frustrating example of difficulty in accessing information, [Bus67, Hub88, Ken78] are all out-of-print and two of these had to be obtained by the time-consuming 'Inter-Library Loan' process.

Some interesting effects emerge when the concepts of compiling and accessing of information are applied to distributed networks themselves (i.e., the subject of a database is the structure and operation of the network).

There can be no doubt that hardware architecture and distributed processing research tackles problems that are germane to searching for information, and will contribute to the achievement of the GyP∫IS Challenge. However, it is inconceivable that architectural research alone will provide all the answers. The cautionary tale about the original attempts to create integrated information systems *within* corporations should make us pause before imagining that the fact that computing power and data storage capacities are adequate is in itself a guarantee of success. At most, architectural research is investigating the nature of possible environments in which the GyP∫IS will have to operate.

An interesting parallel can be drawn with the situation in database technology, in that it is unreasonable to expect a 'best' architecture to emerge for the GyP∫IS. The fact that we need a multiplicity of views of data and information means that alternative forms of 'computation' could prove to be more effective than our currently predominant model. Indeed, in some conventional application areas, alternative architectures, such as neural networks, are already making an impact. It is not even clear that the current dominance of digital techniques will necessarily persist.

3.5 Formal Methods View

The preceding sections present predominantly engineering-oriented views of potential responses to the GyP∫IS Challenge. Here we review the contribution which can be made by more theoretical research.

Abstract data types have received extensive discussion in the theoretical literature, and practical experience confirms that they form an excellent way of encapsulating data so that internal design decisions do not affect the users of such data. However, in the GyP∫IS, it is precisely the need to obtain data from behind a program interface which is the challenge for someone wishing to write a novel program which accesses that data in a way which was not envisaged by the original designer. So, while abstract data types may contribute from the theoretical (and even practical) point of view, it may now be necessary to move on and find more appropriate building blocks for a worldwide information network. In moving on, it is desirable to have the same kind of theoretical underpinning – as was present with abstract data types – for the new concepts which are to be used in future systems.

One natural extension of the abstract data type theme is found in object-oriented concepts (notably [Tok93]). Already, considerable formal methods attention is being paid to such concepts, and it may prove possible to extend them to meet the global requirement of the GyP∫ IS.

Reactive systems are those which go on executing forever and whose behaviour is described in terms of *reaction* to *stimuli* rather than the input-output relation which may be used to specify a simple computation. The GyP∫IS will certainly have to be reactive and, therefore, one could look at the theoretical notations used to describe reactive systems. However, here again, we can see that future systems may have to go beyond what we currently view as a reactive system. In particular, it is difficult to see how anything can respond to the GyP∫IS Challenge if it cannot adapt its behaviour so as to answer queries (about information which is held internally) not in the form which the program was initially specified to handle. Hence, we are facing a class of system in which its interactions cannot be preplanned. There is relevant technical work here in the description of behaviours of systems which cope with mobile processes.

A conclusion from what has been discussed already is that it will be necessary for the system to store contextual information about the environment in which it runs. This is another form of information which will need to respond to interactions with the environment. More speculatively, there needs to be more research on 'semantic structures' and knowledge representation. And a final way in which formal methods may be able to contribute is to tackle the inherent complexity of search in the GyP∫IS.

Many of these topics are discussed elsewhere in this volume, particularly explicitly in the contributions of Robin Milner, on 'Semantical Concepts in Computing', Tony Hoare, on 'Algebra and Models', and Alan Bundy, on 'Prospects for Artificial Intelligence'. Tony Hoare takes the view that theory should be developed more-or-less for its own sake. However, in our view, the thing that formal methods cannot do on its own is to deliver acceptance by the general community of any particular theory. We shall return to this topic again later.

3.6 Summary

There are many complementary technical perspectives to the GyP∫IS Challenge. The ones identified above are by no means the only ones we could have mentioned. For example, as noted earlier, many applications of the GyP∫IS require it to deliver large quantities of information from remote repositories quickly. This raises further research issues, which are associated with the need for 'time-constrained' (more usually known as *real-time*) responses to information queries. There is an existing technology devoted to real-time numerical processing, for on-line process control, signal processing and interpretation (e.g., for radar), and so on, which can certainly contribute to solving the Challenge.

Another important technical aspect is the requirement for meaningful interaction, both among agents, and between human beings and agents. In a companion chapter in this volume, William Newman argues the case for 'Interactive Computing as Mainstream Computer Science', and this is another germane view of the GyP∫IS Challenge.

Our analysis of the technical states of the art suggests that no one of the existing technologies alone can solve all of the problems posed by the Challenge. Nevertheless, each technique has something to offer, and so an 'interdisciplinary' approach may pay off.

4 Pointers to a Response

If our prognosis for the 'information industry' is correct, the GyP∫IS is likely to provide the major source of challenges for informatics research in the coming decades. Hence, in this section, we certainly do not aspire to offer a design which satisfies the challenge set out above. We can, however, attempt to identify at this stage a framework in which such a design could feasibly evolve.

In essence, the preceding sections express dissatisfaction with our present approaches to computer-based management and manipulation of information. As ever, there are two choices for improvement: either we develop a better approach from first principles, using more appropriate theoretical and practical bases (revolution), or we incrementally modify our current approach until it can handle the new challenges (evolution). We rule out revolution on two grounds. Firstly, prior investment is simply too large for the previous approach to be abandoned in its entirety (just think of the problems of 'upgrading' existing repositories – this problem already inhibits people from taking the far easier step of changing from one database technology to another). Secondly, in any case, we are not convinced that a 'better' overall approach can be found.

Hence, we expect that existing information systems are going to evolve in such a way as to meet the needs of global-yet-personal information access. The main strategic objective will be to replace those parts of the overall 'system' that presently rely on human skills by some form of 'computational' activity. Of course, such 'computations' will need to be at least as effective as the human activity they replace. In order to achieve this, they will have to exhibit the new kinds of behaviour described in the remainder of this section.

4.1 Openness

Many systems are deliberately designed as object-oriented programs or abstract data types. In some senses, this is a technical advance over the 'control block' designs of the 1960s, but such abstract data types are designed deliberately to restrict the access to information. We must remember that the GyP∫IS Challenge relies on not pre-judging future modes of access to information, and try to achieve this while at the same time maintaining the discipline of structured design.

One can in fact propose a technical test for an open system. It is possible to decide whether, through interaction with a system, one can acquire (at some cost) enough information to completely reproduce the function of a particular repository (and its moderator). This is not true, for example, of the current machine-readable forms of the Oxford English Dictionary, where one can enquire about the meaning of a particular word, or all of the occurrences of a particular word in other definitions, but not obtain the whole text of the dictionary (except by an exhaustive listing of the entries, which would require prior knowledge of every word that is contained). Of course, open systems can also be abstract data types!

One might ask why the provider of a repository would wish to make available all of the information which it has gathered, perhaps at enormous cost. The answer is economic. In the first place, the acquisition of such detailed information might cost an enquirer so much that they would think twice about building their own version of the repository (and its moderator) rather than using the original one. Moreover, the original provider of the information has the opportunity to hone access paths in a way which makes it practical for them to provide most forms of information access at a lower cost than anyone who simply gets all of the basic entities and relations out of the repository. Finally, the originator is normally best equipped to undertake that the information will be kept up-to-date.

4.2 Adaptability

Many information systems today, whether or not they are officially listed as reactive systems, have extremely restricted protocols for interaction[6]. Not only must protocols be made more flexible, there must be parts of the protocol which are deliberately designed to permit the adoption of new behaviours.

4.3 Broad Search Capability

Remember that the primary motivation is to get as much relevant information as possible from distributed repositories into the hands of the decision makers to some pre-defined schedule.

[6] But we must recognise that the same applies to some (obdurate) human beings.

In many cases, the desire to access information will involve elaborate searches through the repositories. Completing such searches in reasonable time will require sophisticated interactions, via seekers and presenters, between the 'end-users' of the information and the moderators providing the interface to the individual repositories (for the moment, we will assume that repositories are always 'moderated', if only to ensure that access is paid for).

In order to fit into the view of a geographically distributed information network, seekers must be responsive and have the ability both to communicate and to react to incoming stimuli. One could therefore think of a seeker having information which it 'owns' and some form of protocol handler which expresses its possible behaviours during interaction. But, as well as the information which is the *raison d'etre* of the seeker, it will only be able to function if it has an effective (i.e., accurate and helpful) 'environment model' (which is itself information about the interactions which have been found or are expected to yield other sorts of information).

4.4 Mobility

It is our belief that sophisticated interaction, whether among humans, among machines, or between the two groups, demands physical proximity: dialogues with inordinate delays become unmanageable (cf. long-distance telephone calls linked via satellite, as opposed to submarine cable). Also, dialogues with minimal context can easily miss opportunities for 'short-cuts'; for example, a direct conversation with someone contains a considerable amount of 'non-verbal' communication ('body language') that is simply not available over, say, the telephone. Hence, in order to gain the required adaptability, it will be necessary to send the seekers to meet the moderators 'in person'.

We speak here of 'seekers' and 'moderators' and deliberately leave open the question of whether these are embodied in computer systems or whether they are human agents: if an end-user were prepared to pay enough for information that has not yet been gathered, the invoked interactions could ultimately initiate action by a human being who will gather it. However, our aim is to obviate such action if there is any stored source of the sought information.

It is open to debate whether the moderators should build their own internal model of the environment: it must be possible to operate without this. However, efficiency could well be improved if moderators were capable of pointing seekers 'in the right direction'.

4.5 Autonomy

Following from the above, seekers must be capable of operating autonomously, or at least semi-autonomously. First and foremost, each seeker will need to operate with some 'sense of purpose' (cf. the comment from Brian Gladman, cited in Section 3.2), in a context that is 'inherited' from the end-user it is serving. Because it will be mobile and frequently away from its 'base', a seeker will at least need to 'touch base' every now and then for instructions, either 'in person' or via a (cheap) telecommunications link. More likely, for reasons of efficiency and cost, it will have to operate for prolonged periods without feedback from its end-user.

4.6 Economic and Social Dimension

There seems no reason to doubt that the information itself can be represented in something like the entity-relation view. Seekers, then, are involved in interactions whose behaviour is mediated by the protocols which they can conduct. In addition to the sought information, there must be economic information which indicates both the cost of, and time required for, access (searching, processing and moving) and, possibly, some knowledge of the constraints on access required by the law. Whenever a task is to be undertaken, some form of price will be established and thresholds set on queries or searches sent to other agents within the GyP∫IS.

Economic incentives will also serve to improve the general capabilities of the GyP∫IS. Repositories that do not adhere to the open philosophy will be vulnerable to competition from more market-minded systems: they must evolve to meet the demands of their 'customers', otherwise they will 'go out of business'.

4.7 Summary

Although the totality of these behaviours represents a big change in approach from current practices, no single step is frighteningly large. This would seem to support our contention that evolution towards the objectives of the GyP∫IS Challenge is at least conceivable.

5 On the Nature of Informatics

There is a danger that the reader could view the foregoing, especially section 3, as suggesting that all current computing research is germane, and that we should all carry on working the way we already are. But this is not the case. Quite apart from the need to focus on critically selected key research issues (brought upon us by the inexorable tightening of research budgets), we see a 'bigger picture', in which underlying problems will increasingly manifest themselves unless the informatics research community takes specific steps to alleviate them. To help the reader understand this view, we have added the following 'epilogue', on the nature of research in the emerging discipline of informatics.

5.1 Informatics as Science

Science is the process of gaining knowledge and understanding by means of methodical elaboration. The objective of scientific method is the construction of increasingly effective models (effective in the sense of facilitating ever more accurate *prediction*).

In informatics, we wish to understand the 'world around us' by constructing increasingly effective *information-based* models. The method we use for this follows the usual cycle of hypothesis (creation of a model) and test (using the model in some planned experiment against the 'reality' and evaluating its validity, or otherwise) that characterises the physical sciences. Many of the other contributions to this volume describe how this kind of cycle has driven the evolution of particular specialities in the computing field; and these all make the point that it is the science of the field that is being advanced, not just the technology.

Note that non-trivial testing requires a substantial amount of high quality engineering, to construct the necessary experimental apparatus, as well as the observational and analytical skills more often associated with experimental science. This point is amplified in the chapter by Simon Peyton-Jones elsewhere in this volume.

In common with the physical sciences, there are tensions between the groups that specialise in modelling (the 'theorists') and those that specialise in testing (the 'experimentalists'). There are also tensions between different 'theories' (and, hence, between the theorists). The latter are due to the vast number of ways there are of modelling 'reality', and the former are due to the frequent discoveries of inconvenient incompatibilities between modelling techniques and ‘reality’ (as well as the fundamentally different viewpoints inherent in the two groups)[7].

Also in common with other sciences, informatics has started by investigating simple phenomena, trying to understand small, bounded (closed) systems, with few real-time constraints, before expanding into the more complex areas implied in the GyP∫IS Challenge.

5.2 Historical Influences

Of course, informatics is an *emerging* science, and not everyone who is working in it necessarily subscribes to the overview presented above. Perhaps because they do not appreciate the existence of the tensions identified above, people argue about the relative merits of different models or experiments more ‘blindly’ than they would in a more established science. After all, physics and chemistry have had several centuries of evolution in which to ‘hammer out’ their differences and develop a commonly agreed methodology for advancing their discipline. Whatever the reason, it is certainly the case that self-damaging competition and disparagement has been rife in certain prototypical areas of informatics (e.g., computer science).

In our opinion, it is much more fruitful to note how the differing approaches connect to one another and to related fields of science. In essence, we ought to be getting on with the ‘hammering out’ process so that we can agree a methodology for advancing our discipline. And analysing our ‘roots’, in the various fields of computer science and elsewhere, is one of the essential parts of that process which helps to illuminate our collective scientific ‘agenda’. At one level, this was the motivation for surveying possible approaches to the GyP∫IS Challenge in Section 3.

In fact, one can see from Section 3 that informatics is more than ‘a science’: for example, a great deal of its intellectual heritage belongs more naturally to the collective discipline of engineering. So, we find discussion of an issue such as information processing in machines alongside comparisons with the way that human beings handle information; or a communications engineer might wish to look at the way data flows through communication media such as wires or light pipes; while a social engineer might be interested in the ways in which humans react to the information they receive. There is certainly a prevalent engineering view in informatics: many people wish to build systems that work and work efficiently. There is another view which is like that of an experimental scientist: some people wish to make observations, develop models, and test whether those models are actually embodied in information systems. And there are the mathematically-oriented informatics researchers who are interested in the abstract theoretical structure which underpins their subject. The latter aim to provide the structure of an evolving science, and the recent publication of several series of ‘Handbooks’ which collect and record basic theoretical material is the first instalment of this promise.

Viewed from a suitable distance, all of these research activities can be seen to contribute to the scientific rationale offered above. The great variety of approaches available is a huge asset; there is the potential to integrate diverse viewpoints and create a totality that far exceeds the sum of its parts[8].

The major question from a methodological point of view is how and when to deploy each of the various skills available to the advancement of the discipline. Petty wrangling simply delays

[7] And this is quite apart from the tensions between these academic viewpoints and the views of industry and commerce that so dominate present-day research funding strategy. The potential negative effects of such tensions is amply illustrated in Roger Needham’s chapter in this volume.

[8] Those readers who know the authors will realise that our backgrounds are very different. The fact that we are both able to subscribe wholeheartedly to this scientific rationale is significant.

progress, so the lesson to be learned from history is to cooperate more effectively.

5.3 Synthesis

With this background in mind, we propose that progress be made by deliberately looking at the GyP∫IS Challenge from successive, fundamentally different viewpoints, with the aim of finding the particular view that enables us to make an advance at the moment in question.

So, for example, computer architecture, programming and database design can be viewed most straightforwardly as engineering disciplines. Programs (and hardware systems) have to meet specifications and an important aspect of a programmer's task is to come up with an efficient product which satisfies the given specification. There is of course much established technology here. The notions of abstract data types, object-oriented languages and expert systems are all relevant to the sort of programming tasks needed in the GyP∫IS. What is important in facing the GyP∫IS Challenge is to carry forward the sort of engineering view which recognises that both ease of use and efficiency are vital considerations.

Elsewhere in this volume, Alan Bundy makes a strong case for present-day AI to be thought of as an experimental science. Viewing the GyP∫IS from this perspective, one might concentrate on the anthropomorphic qualities of the seekers and presenters, and on the 'fitness' of successive versions for service in 'society'.

What formal methods have to offer is the ability to model or describe complex computer systems and to reason about their implementation in a way that can, for example, show that design decisions satisfy their specifications. They do not necessarily solve the engineering problem of finding an efficient solution. The more important contribution of the formal methods point of view is that it can give analytic tools to designers of systems that allow them to simplify their architectures.

There is a major challenge in formulating the specification of the system(s) needed. The GyP∫IS contains information about the physical world. A physical scientist is concerned with building and animating *models* of physical reality, with as much veracity as possible. In order to avoid the pitfalls of earlier information systems (inaccuracy, incompatibility, inaccessibility, etc.), it is mandatory that the connection between physical and abstract world be adequate. In particular, the connection between the GyP∫IS and *its* model must be 'watertight'.

No one of these viewpoints is paramount – what is interesting is how an optimal solution can only be approached by balancing the tension between their diverse perspectives. In particular, their must be agreement across the representatives of each viewpoint on the nature of any compromises that have to be struck. Other contributions to this volume amply illustrate that progress in some fields has been made by shifting the collective viewpoint: take, for example, Alan Bundy's (slightly defensive) admission that the AI field spent much of its 'youth' in what he calls a 'scruffy' state of 'exploratory programming' before emerging into its present, 'respectable' persona. It is quite probable that the early state *had* to be endured in order to overcome the specific problems then being encountered.

Perhaps the penultimate word can be left to Keith Devlin [Dev91] who writes:

> That there is such a thing as *information* cannot be disputed, can it? After all, our very lives depend upon it, upon its gathering, storage, manipulation, transmission, security and so on. Huge amounts of money change hands in exchange for information. ...But *what* exactly is it? The difficulty in trying to find an answer to this question lies in the absence of an agreed, underlying theory on which to base an acceptable definition.

If the ‘agreed, underlying theory’ is to be found, it will not be by theorists, or engineers, or experimental scientists alone. Notwithstanding the tensions between them, they must all work consciously together towards the discipline’s common goal.

Acknowledgements

The authors are grateful for comments on earlier drafts of this article from Tim Clement, Ian Hayes, Robin Milner and Dave Snelling. One of the authors attended the *Software-2000* Workshop and wishes to express thanks to the attendees for stimulating some last-minute examples which have been added (the report of this meeting – [RRW94] – emphasises the key role of information to the future of the software industry).

References

[AMR94] Peter Aiken, Alice Muntz, and Russ Richards. DoD legacy systems. *Communications of the ACM*, 37(5):26–41, 1994.

[BDMS94] C. Mic Bowman, Peter B. Danzig, Udi Manber, and Michael F. Schwartz. Scalable Internet resource discovery: Research problems and approaches. *Communications of the ACM*, 37(8):98–107, 1994.

[Bus67] Vannevar Bush. *Science is not enough.* William Morrow, 1967.

[CAB+94] Derek Coleman, Patrick Arnold, Stephanie Bodoff, Chris Dollin, Helena Gilchrist, Fiona Hayes, and Paul Jeremes. *Object-Oriented Development: the Fusion Method.* Prentice Hall, 1994.

[Dev91] Keith Devlin. *Logic and Information.* Cambridge University Press, 1991.

[DLC89] Edmund H. Durfee, Victor R. Lesser, and Daniel D. Corkill. Cooperative distributed problem solving. In Avron Barr, Paul R. Cohen, and Edward A. Feigenbaum, editors, *The Handbook of Artificial Intelligence IV*, chapter XVII, pages 85–147. Addison Wesley, 1989.

[Fre91] Karen A. Frenkel. The Human Genome Project and Informatics. *Communications of the ACM*, 34(11):41–51, 1991.

[GL94] R. V. Guha and Douglas B. Lenat. Enabling agents to work together. *Communications of the ACM*, 37(7):127–142, 1994.

[HdJ84] Carl Hewitt and Peter de Jong. Open systems. In Michael L. Brodie, John Mylopoulos, and Joachim W. Schmidt, editors, *On Conceptual Modelling*, chapter 6, pages 147–164. Springer-Verlag, 1984.

[HK91] A. Sharif Heger and Biily V. Koen. KNOWBOT: an adaptive data base interface. *Nuclear Science and Engineering*, 107:142–157, 1991.

[Hub88] B. A. Huberman. *The Ecology of Computation.* North-Holland, 1988.

[Jac83] Michael Jackson. *System Design.* Prentice-Hall International, 1983.

[Ken78] William Kent. *Data and Reality.* North-Holland Publishing Company, 1978.

[LB85] M. M. Lehman and L. A. Belady. *Program Evolution: Processes of Software Change.* Academic Press, 1985.

[McL62] Marshall McLuhan. *The Gutenberg Galaxy.* Routledge and Kegan Paul Ltd, 1962.

[MGT+87] Thomas W. Malone, Kenneth R. Grant, Franklyn A. Turbak, Stephen A. Brobst, and Michael D. Cohen. Intelligent information-sharing systems. *Communications of the ACM*, 30(5):390–402, 1987.

[PB94] William J. Premerlani and Michael R. Blaha. An approach to reverse engineering for relational databases. *Communications of the ACM*, 37(5):42–49, 1994.

[RBP+91] James Rumbaugh, Michael Blaha, William Premerlani, Frederick Eddy, and William Lorensen. *Object-Oriented Modelling and Design.* Prentice Hall, 1991.

[Rie94] Doug Riecken. A conversation with Marvin Minsky about agents. *Communications of the ACM*, 37(7):23–29, 1994.

[RRW94] Brian Randell, Gill Ringland, and Bill Wulf. Software 2000, 1994.

[Sim69] Herbert A. Simon. *The Sciences of the Artificial.* MIT Press, 1969.

[Tok90] M. Tokoro. Computational field model. In *Proceedings 2nd IEEE Workshop on Future Trends in Distributed Computing Systems*, 1990.

[Tok93] M. Tokoro. The society of objects. In *OOPSLA'93 Proceedings*, 1993.

[Wal94] Paul Wallich. Wire pirates. *Scientific American*, pages 72–80, March 1994.

[WWC92] Gio Wiederhold, Peter Wegner, and Stefano Ceri. Toward megaprogramming. *Communications of the ACM*, 35(11):89–99, 1992.