SWP 49/87 OPEN SYSTEM CONSTRUCTION: A STRATEGY FOR TURBULENT TIMES

ANDREW J. BYTHEWAY
Centre for Logistics and Transportation
Cranfield School of Management
Cranfield University
Cranfield
Bedford MK43 0AL
United Kingdom

Tel: +44 (0)234 751122
Fax: +44 (0)234 781806

OPEN SYSTEM CONSTRUCTION - A STRATEGY FOR TURBULENT TIMES

Andrew J. Bytheway

Cranfield School of Management

A paper presented at the British Computer Society Database Specialist Group Conference Edinburgh 1987

Abstract:

The rapid rate of change in information technology compounds with commercial pressures in a way that requires information systems to be implemented quickly and effectively. This can be critically important to the competitive advantage of a business enterprise. Productivity in system development is not enough, however. Development effort has to be well directed and at first sight the new ideas for strategic analysis of information systems are very attractive.

By drawing upon a real case study where strategic analysis techniques were used it is possible to show that strategic analysis requires its own discipline if it is to succeed, and that it will lead to new pressures on the system development process. The mixing of bespoke, packaged and reusable system components from different sources will become an important factor in implementation, and what we know as system development will become more a process of system construction. If this is to be possible a new "open" approach to the whole process will be required.

There is a new international standards initiative which should make an important contribution to this problem. Where the OSI reference model provided a framework for open system interconnection, it is hoped that a new reference model for system development will provide a framework for open system construction (OSC).

Introduction

These are difficult times for those who are concerned with information system development. Skills are short and application backlogs are growing. The situation in any enterprise that is dependent upon computers is likely to be worrying. There will be several different existing computer systems which support the routine operations of$the business and which are based upon a great deal of existing software. Much of the software will have been written some time ago and it may or may not have been documented - it is probably not now understood by anyone in any detail.

Organisations are under pressure to become more efficient, and (in the commercial case) to become more competitive and more attuned to the needs of its customers. Senior management will be more aware of the potential of information systems to make a contribution to the business strategy, and the mystique which has previously allowed the data processing department to direct its own future is no longer acceptable.

In these circumstances DP management has to develop a strategy for survival. Bringing in new systems is becoming more difficult and the newer technologies such as fourth generation languages and relational databases introduce as many problems as they solve, it sometimes seems. It is probable that some parts of new systems will be best satisfied using a package where others may be already satsified in large part by existing applications software. Yet other parts of new systems that are closely aligned to the company's competitive strategy are likely to have to be specially written: these systems must be "better" than those used by the competition. Overall, in order to meet the corporate need in the best way it will probably be necessary to integrate of a mixture of packaged, existing and bespoke software.

Even if we are able to develop strategies for information systems which help us to identify these parts, the tactical and logistical problems remain. The tasks within the total development activity (and the people who achieve them) must all be incorporated in an overall plan. The bespoke software may provided by one software house and packages come from others. Consultants may be brought in to assist in the business analysis and requirement definition. Regulatory bodies might be involved, and there may have to be new equipment and other supporting services to be bought.

Because of the complexity of bringing all these disparate parts and participants together, the existing installation standards will surely prove woefully inadequate. The partners involved - the in-house DP staff, equipment suppliers, software suppliers, software houses, consultants, users, regulatory bodies - are certain to have dramatically different views of the nature of standards and their relevance. Some will have methodologies and some 4GLs. Some will talk of quality control and some of configuration management, Some will have workbenches and some may talk of IPSEs. Some partners will not be understood and others will not understand. All in all we are faced with a nightmare of difficulty and potential disaster.

We need to find a defence against the technical and commercial turbulence that surrounds us. If we stand back from the problems and reflect upon them from a proper perspective we might come up with a "wish list" of those things that we would like to help us straighten out our affairs:

- a more productive way of implementing systems;
- a proper strategy for sorting out and prioritising the backlog of systems development;
- better ways of integrating new and old systems, and of avoiding doing our jobs twice over;
- better ways of matching our requirements to the capability of packages.

Productivity

The simple view has always been that if productivity is improved most of our problems will go away. This argument does not stand close inspection however, and it raises more questions than it answers.

Programmer productivity:

Increased programmer productivity is an attractive target and can be approached in a strictly clinical and quantitative way - "#4.50 per line of corrected code" ... "2000-3000 lines of working code per programmer year". Programmer productivity can indeed be improved but with only limited overall benefit. Whether it helps to reduce the applications backlog is questionable, and if we are to measure overall productivity we must adopt a much broader perspective. For example: what about the re-programming that derives from misunderstandings at the analysis stage? What about the optimisation that has to be done because the system is not fast enough? What of the code that is written three or four times over because nobody stopped to identify and separately specify the common modules? Is this all included in our concept of productive code? If so, what proportion is it of the whole?

Detailed analysis of the number of source lines written and their ultimate disposition confirms our worst fears. One analysis of source lines written in large projects [Youll 1984] showed an alarming degree of wastage:

paid for but not delivered:	28%
delivered but not used:	48%
used and abandoned:	19%
used with changes:	3%
used as delivered:	2%

If these figures are representative it is clear that saving a small proportion of programming time is nothing compared with getting the specification right in the first instance. The figures are a reflection of the rule that one hour of lost analysis time can cost many hours of design and programming time. Most of us today retain some interest in programmer productivity, but are equally interested in the quality an productivity of the analysis and design work that precedes programming.

Analysis and design productivity:

Against the background of wasted programming effort it is true that systems analysis and design techniques are changing in order to reduce the programming waste. Structured methods are being quite widely adopted, but in the majority of cases their introduction <u>extends</u> rather than reduces the initial elapsed time in a project. The justification for using them is that the time needed for programming will be much less, and overall there will still be a saving. It is not yet clear whether the overall effect is actually beneficial in the majority of cases, and in some notable cases there are clearly problems [Collins 1987].

Because of the intensity of effort involved in the use of structured methods the productivity of the "structured analyst" has become a primary issue. It has become clear that machine-based aids for the analyst are helpful and the analyst/designer workbench has emerged. There are high hopes for the productivity of these workbenches but there are no clear patterns yet. One early (but detailed) analysis [Bytheway 1984] was based on experience with a prototype workbench. It mapped the capability of workbenches onto well- defined tasks within the overall development cycle (and ancilliary activities) and produced the following estimate of the percentage saving in different tasks:

Feasibility	27%
Analysis	28%
Design	24%
Implementation	17%
Documentation	29%
Training	7%
Auditing (QA)	13%
Overall weighted average:	19%

At the time this provided no more that a target to aim at, but subsequent experience with proprietary workbenches is showing that it is <u>not</u> achievable in an easy and consistent way.

And so we find that programmer productivity is only one factor, and that analyst and designer productivity may be much more important. As there is wastage in the work done by programmers, so we must expect that there will be wastage in the sork done by analysts and designers. Perhaps this is avoidable if we get the overall strategy for information system development right? The idea that productivity is a broad issue is not new [Gilb 1983] but the current trend to review information systems from a business strategy viewpoint is [Wyman 1985].

The strategic view of productivity:

To the end-user productivity judgements are based almost entirely upon cost and elapsed time, perhaps tempered with some consideration of quality. Given the opportunity to inspect the productivity of the system development process himself his criteria would be much more concerned with the relevance of each project to the overall business strategy, and he would find that it is not always satisfactory. Where there have been such audits as this, it has been found that only a minority of the development effort is relevant to current business strategy.

Thus the productivity argument is subsumed into an argument for a proper strategy for system development, and for the classification of application systems development according to the strategic needs of the business.

The boundary between business strategy and information systems strategy is a most interesting one. Conventional management ideas about markets and products help us to take a more appropriate view of the development task. Indeed, the notion that we might be able to implement systems in different ways and at a rate that varies according to the nature of the system and the attitudes of the user can be formalised very easily.

A Scheme For The Classification Of Application Systems

Conventional marketing theory gives us a way of classifying systems. The classification of markets or products is usually done according to criteria such as the resource cost and the commercial benefits - we can employ these criteria in our domain.

New markets can be very expensive to operate in. Until the potential is proven there will be little impact on the organisation and little revenue benefit - they have to be "turned round" first. If they are turned round and there is benefit they have to be addressed in a much more disciplined way - they are in effect "strategic" markets that have to be integrated into the enterprise, and which will begin to earn significant revenues. Once brought under control the strategic market must be serviced in the most cost effective way so that the benefit of the revenue can be maximised. This is usually referred to as the "factory" mode, implying high volumes, efficiency of production, and (hopefully) high profits. Eventually, the potentional of a market will become exhausted and it has to be tolerated only in order to service existing customers and fulfill long term obligations.

This can be summarised in a two-by-two matrix:

	Great benefit to enterprise	Low benefit to enterprise
High resource cost	New markets with proven potential "Strategic mode"	New markets with unproven potential "Turnround mode"
Low resource cost	Well established profitable markets "Factory mode"	Old markets that are spent - "Support mode"

In effect this model is a life-cycle for markets: they tend to migrate around the matrix (in an anti-clockwise direction) as they mature.

A comparable model has been proposed to assess the strategic impact of information systems in an enterprise [McFarlan 1984]. The original model can be evolved in a way which helps to classify systems very easily:

	High impact on enterprise	Low impact on enterprise
High resource cost	New systems with proven benefits "Strategic mode"	New systems with unproven benefits "Turnround mode
Low resource cost	"Stable, high usage systems "Factory mode"	Old systems that have to be kept "Support mode"

o Turnround systems:

"Turnround" systems are those which are user-driven, experimental, emerging, and probably ill-formed. They will not generally be closely related to existing systems, although the time may come when they have to be integrated.

They will be expensive in the sense that they will tie up expensive resources and provide little overall benefit. Their impact upon the organisation is very low, and they will have little immediate strategic significance.

o Strategic systems:

Successful turnround systems will be recognised as having significant potential in the enterprise. They then become strategically important and have to be brought under a degree of control - in effect they become "strategic" systems that will be helping the enterprise to move forward.

They will have to be integrated with existing systems, and (although the cost will be high even where they are successfully integrated) they will begin to have a significant and beneficial impact on the organisation.

o Operational systems:

Strategic systems which are stable, integrated and well- understood become "operational" systems. They have to operate efficiently at high volumes and are central to the execution of the business of the enterprise.

It is here that information systems make their greatest contribution. Their costs should be low (or at least well controlled), and the benefits are reflected in the primary revenue-earning activities in the enterprise.

o Support systems:

Those systems that are supporting spent markets or products, or which are perhaps necessary for some reason not connected with the central business objectives (for example to satisfy statutory requirements) are "support" systems.

They have to be maintained, but their contribution to the enterprise is slight. The costs ought to be minimised.

In discussing IS strategy this classification scheme is very helpful indeed. It clarifies many problems such as: the way in which the overall development workload can be best prioritised, the best balance of resource allocation, the style of project management best suited to different projects and the technical strategies that are most likely to achieve efficiently the different applications.

For example an excessively prescriptive style of development (with rigid standards and pre-defined milestones and review points) is wholly inappropriate for turn-around systems. It would stifle experimentation and prevent the rapid development of new ideas which might be critically important to the future - better perhaps to use a 4GL without any project controls at all. On the other hand, a lack of discipline in the development of high volume systems which have to support central corporate activities would be disastrous - these systems must be well integrated and efficient in execution.

A Case Study

But how do these strategic ideas hold up in practice? Do they lead to a more productive implementation of systems? There are few published reports of studies which have employed them, although there are signs of activity amongst the consultancy companies that operate in these areas [Sweet 1987].

The case study presented here is not a large one, but it is real and it is interesting. A small team of consultants had developed methods for the identification and refinement of information system strategies, with help from the Cranfield School of Management. The ideas had been applied in a limited sense on other projects, but not in any complete way. The opportunity arose to go "live" with a client who had all the right qualifications - very real problems, a considerable will to surmount them and great potential for the beneficial application of information systems. The study provides a much better understanding of the consequences of a strategic approach.

The company:

If one believes that there might be a renaissance in British industry, then the company in question exemplifies the problems and opportunities involved. It has a recent history based upon the manufacture of electromechanical products for the control of process plant - mainly eddy current couplings. The manufacture of these analogue devices requires a lot of metal working, they take up a lot of space, and they are primitive in their actual control capability. They might typically be used to control rotating machinery, large fans or other electrically driven equipment. It is clear that digital control systems are likely to provide vastly improved control, and other indirect benefits.

The new product line:

The microprocessor is widely used in the new products which have been designed to replace the old ECCs. The company was able to recruit a team of young engineers and designers, having had the benefit of the largest microprocessor application development grants ever awarded by the British government.

There is much more to it all than just the microprocessor, however. The company has always sold standard control products, as if from a catalogue. In this position, where cost is probably the main buying citerion and where volume production is critical to the control of costs and the preservation of margins, any small company is going to be at great risk from aggressive competitors. There is much to be gained by a company under this kind of threat if it can liberate itself from supplier dependencies and increase customer dependencies [Porter 1979]. The strategy adopted was centred on a change to the supply of systems rather than products, where most of the added value comes from engineering design skills rather than the manufacturing process: the customer is much more dependent on the supplier of a complete control system than he would be on the supplier of components for such a system. Digital control systems can, to a large extent, be built from easily stocked components that are available from multiple sources. The competitive threat is diluted and buyer switching costs can be raised.

The need for new information systems:

However, if complete control systems are to be tendered, engineered and commissioned quickly and reliably, then the availability of information about the make-up of those systems becomes critical. The components are now largely digital rather than analogue, and they include software as well as hardware. Information about component specifications, re- usability, cost, changes, alternatives, supersessions and sources of supply must be easily available to support the sales process, the design process and of course the manufacturing process. The total number of components will be much higher than with the old analogue products, and there will be a complex hierarchy of elementary parts, printed circuit boards, sub-assemblies and completed assemblies in each system, instead of just a few drawings. The need to analyse the impact of changes will be paramount. Most important, the need to have information available in an integrated way, across organisational boundaries and without duplication, quickly becomes apparent. The sales effort must succeed in selling something unique and complex, that the production facilities can produce without significant special engineering. The materials management processes must ensure that the right components are available on a timely basis.

But we are rushing too quickly into the detail of the project as it turned out, which is not our main purpose here. The study was an important component in the development of the business, because whilst some cash was taken care of by the government grant, there is an American parent company that also has to be convinced. It is a large conglomerate that is cash-rich and complementary in terms of product and skills. Despite the apparent availability of further cash, nothing comes for nothing, of course. Clear plans have to be put on the table describing the further development of new products and of the business itself before additional finance will be forthcoming from any source.

Purpose of the study:

The purpose of the study was to document the existing systems, to identify the need for new or amended systems, and to present strategic system development options on a two-year timescale, with some indication of the resourcing and costs involved. This was to form a major part of the business plan and provide evidence of the need for investment.

Gathering the facts:

The initial part of the study was a classic pattern of fact finding. The four-man senior management team was interviewed, and this lead to follow-up interviews with a small number of chosen line managers. The findings were reviewed with senior management. The process of discovery, where one begins to hear things which initially seem disjoint, and which then come together, and ultimately begin to be repeated, indicated that the first cycle of fact finding was nearing completion.

The information systems portfolio:

The portfolio of computing facilities, in terms of hardware and software, was rather haphazard and based more upon historical accident than careful planning. Having said that, there seemed to be little computing activity that was unnecessary and a good deal that was appropriate. A major element was the use of a bureau facility with which the company had once had a strong connection. The connection was no longer, and it was worrying to find that the bureau was using a non-standard version of a well-known IBM software package for materials management, and that the likely level of support available was seriously limited. This bureau facility was clearly a supplier dependency that the company could well do without.

Information processing outside the scope of the bureau service was as much manual as computerised, but with increasing use of personal computer software. As many other organisations have found, this was both good and bad: PCs liberate the user from the shackles of external dependencies, but at the same time they can become exhausting in their demands upon the user's time. Systems developed on PCs tend to duplicate data held elsewhere, and in the worst case there is a new kind of dependency upon one's own commitment to the PC.

In addition to the essentially administrative systems that were found, there was a technical software development facility used in product development. This was quite specialised and had no immediately apparent connection with the other systems.

Stated and perceived objectives

It is necessary to be very clear about the objectives that are expressed by the higher levels of an organisation, and the way in which they are perceived lower down in the organisation. The popular notion to discriminate between "what must be done" and "how it is to be done" is actually a very relative thing - it depends upon the viewpoint (in the organisational sense) that is adopted in any particular case. We shall return to this theme later. For the current purpose of the exercise, objectives were classed in three ways: corporate objectives, business objectives and information systems objectives

Critical success factors:

At all points where it seemed appropriate, interviewees were asked for amplification of their objectives in the form of critical success factors. There is no purpose in setting objectives which are un-measureable or which can not be supported by criteria for success. Most importantly, the discussion of CSFs tests the clarity of the interviewee's thinking, their true resolve to achieve what they say, and their general comprehension of what they are talking about. It was very interesting and very valuable as a technique.

Problems:

No analysis would be credible if it did not incorporate the rag-bag of problems that always come out during interviews. Without express provision for the recording of problems, many important indicators can be lost which are a valuable part of the analysis.

Analysis:

In this way the fact finding covered all the basic ground. It was based on a limited number of interviews by two consultants, and it produced of the order of 200 individual, apparently "atomic" notes - each being in effect one or another kind of objective, a CSF or a problem. After collation and distillation this produced:

- 6 essential corporate objectives
- 8 distinct additional business objectives
- 14 information system objectives
- 11 significant problems

that is, 39 items. The reduction of the raw notes into this refined form was a non-trivial process. Even with the assistance of data management software to edit and sort the notes it occupied a considerable proportion of the project schedule.

Let us remind ourselves that we are dealing with a real but small company, and that the study was confined to a first pass at the strategic requirement. In another case it might be necessary to have many more interviews and the work might be done by a sizeable team of analysts. In these circumstances the viability of the strategic analysis process could be seriously threatened. The volume of notes can be anticipated, but the time needed to deal with them can be surprisingly high.

This was not the only problem faced. In support of the analysis of objectives and problems, the analysis of the business activities and their classification into turnround, strategic, operational and support activities was not straightforward. The ideas which seem so obvious in the classroom were suddenly inappropriate or inadequate, it seemed.

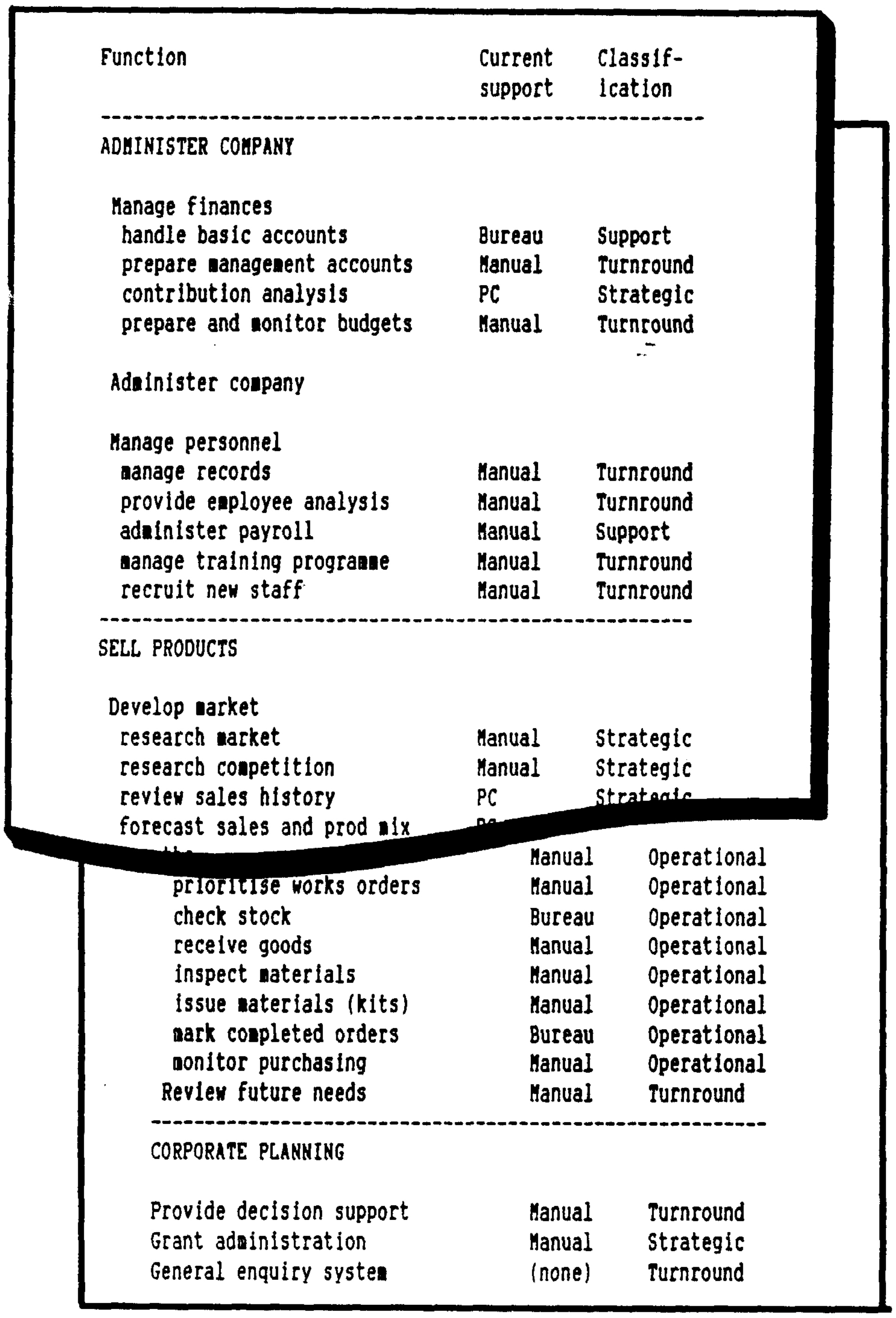
The business activities in detail:

The study identified of the order of 80 functions within the business (see the table below). Deciding where to stop in decomposing the activities was of course a problem. Decisions were taken not according to likely packages or manageable subsystems, but according to the view that was taken by the management interviewed. It represented their view of their needs, not the consultants' views nor the technical opportunity. It follows that some headings are natural candidates for a package, some are yet to be defined in more detail, and some will probably be grouped together and consolidated into a single heading for system development.

The table shows a selection of the identified functions, and indicates how they were being supported at the time of the study and what the agreed classification of them was.

It can be seen that within a single main functional area there are different kinds of system, and that the turnround, strategic, and other classifications follow a pattern but are not uniform within a main heading.

The intention at this point is that the required systems are sorted according to their classification, and according to other characteristics that are not included in the table above. This provides a basis for determining the policy towards information systems development (ISD), and ensures that it is optimally geared to the real business requirement rather than to any particular hardware and software opportunities that may be perceived, and also independently of the short term pressures that tend to distort a less well- considered view.



Function	Current support	Classif-ication
ADMINISTER COMPANY		
Manage finances		
handle basic accounts	Bureau	Support
prepare management accounts	Manual	Turnround
contribution analysis	PC	Strategic
prepare and monitor budgets	Manual	Turnround
Administer company		
Manage personnel		
manage records	Manual	Turnround
provide employee analysis	Manual	Turnround
administer payroll	Manual	Support
manage training programme	Manual	Turnround
recruit new staff	Manual	Turnround
SELL PRODUCTS		
Develop market		
research market	Manual	Strategic
research competition	Manual	Strategic
review sales history	PC	Strategic
forecast sales and prod mix	[illegible]	
[illegible]	Manual	Operational
prioritise works orders	Manual	Operational
check stock	Bureau	Operational
receive goods	Manual	Operational
inspect materials	Manual	Operational
issue materials (kits)	Manual	Operational
mark completed orders	Bureau	Operational
monitor purchasing	Manual	Operational
Review future needs	Manual	Turnround
CORPORATE PLANNING		
Provide decision support	Manual	Turnround
Grant administration	Manual	Strategic
General enquiry system	(none)	Turnround

TABLE OF IDENTIFIED BUSINESS FUNCTIONS (Sample only)

Information systems which support turn-around activity should be rapidly developed in a way that leaves them flexible, even if at the cost of efficiency (for example with 4GLs). Perhaps the user can be involved and reduce the demand upon ISD resources. Strategic systems are very demanding on management and ISD resources, they create unwelcome changes in existing practice, and they create conflicting priorities - there is a limit to the amount of strategic development that can be supported and tolerated at one moment in time. They will probably have to be bespoke systems, centrally developed. Operational systems have to be highly efficient and totally reliable: they might be better implemented with packages (if a reliable and efficient package can actually be found ...) or again by bespoke programming. Support systems do not warrant any avoidable investment, and packages are to be preferred.

This is an admirable intention, of course. In practice it is not quite like this.

The Problem of Granularity

The objective is to identify appropriate, legitimate and beneficial systems for development that will support the corporate strategy. In the study everything had gone well: the interviewees had been co-operative, the corporate strategy was perfectly clear and almost all of the usual problems were absent. The analysis had produced a list of candidate activities requiring supporting systems, but the discipline of the analysis was threatened by a number of things:

- The activities are actually elusive and can always be divided into smaller parts. A part of one system is shared by another, and one person's view is rarely pitched at the same level as another person's view. The most appropriate degree of detail is not easily identified.

- The volume of data in the analysis had become very high, and the absence of a workbench to support the strategic analysis was making it very hard work - even though all the notes and information had been loaded into a data management package. The granularity of the initial analysis had already produced a large volume of information that was at the limits of the capability of "ad hoc" analysis.

- The targets were all moving, and the closer one looked the more apparent it was. Between two interviews the PC systems evolved significantly - for example suddenly a link was in place to the USA where there had been none! The granularity of the analysis had become so fine that the problem was beginning to look like quicksand.

- The appealing idea that a turnround activity becomes (in some orderly way) a strategic and then later an operational one is just not that easy. The various parts of an activity were found to be in different classifications, and the classifications were found to be changing within the timescales of the study. They were both relative and interdependent - changing one seemed to affect another.

The whole exercise was at risk of becoming a nightmare. With doubt as to the level of precision required and repeated difficulties in deciding what to decompose into parts, it became clear that a discipline of some kind was needed.

The concept of granularity, and its use in controlling these kinds of problems is important in achieving discipline. The challenge is to manage one's level of attention in the analysis so that it is appropriate in a specific case and yet coherent in the overall case. It is the same challenge that is faced by the systems analyst, and which has been addressed in the techniques of structured systems analysis.

Granularity in systems analysis

It is often argued that the analyst may analyse from the bottom up or from the top down; in the former case the lowest level of detail that is appropriate is effectively self- defining (the "bottom" is where one starts), but in the latter case there is a problem: in using top down decomposition techniques the most difficult things is learning when to stop, just as was found in the strategy study.

The literature refers sometimes to the concept of "what" and "how". "What" is done is that we "receive delivered goods" (say). "How" we do it is: "first look at the label, second find the packing note, third find the delivery advice, fourth open the box ...." and so on.

This is a very relative thing, of course, as we have already noted. One man's "what" is another man's "how". The warehouse manager has a "what" that is maintaining the stock levels; "how" it is done is by ordering regular deliveries. The computer terminal that supports the receipt of delivered goods has a "what" that is looking up the related order details; to the goods inwards staff this is just a "how" - a detail within the receiving procedure. We need to have better rules to guide us as to the degree of granularity that is appropriate in models of systems that are evolving.

Philosophically this is the same problem as the definition of "real time". In one context it is one thing, in another it is quite different. A guided weapon control system requires a granularity of time that deals in microseconds; real time in this case is a timescale that discriminates between a few metres or minutes of arc in the trajectory. A mail order system requires a granularity of time that deals in days and weeks; real time here is anything that can discriminate between delivery in three weeks rather than four weeks, perhaps. There are other dimensions to granularity which are concerned with the precision with which we specify data, and the increments with which we progress the function in a system.

Rather than refer again to the what/how distinction, it is more constructive to argue that the level of detail must be appropriate to the circumstances, and that decomposition is complete when we have derived detail which suits our purpose at any moment in time. For example, when the detail distinguishes between the existing system and the proposed system, during the earlier stages of analysis; or when the detail distinguishes between clerical and computer processes at a later stage; or in a strategic study when the detail distinguishes the managing director's view from the financial director's.

In any non-trivial analysis there can be considerable difficulty in dealing with this problem. If the work done by one analyst (who was tasked with one subsystem) is to be reconciled with the work done by another analyst (who did another subsystem) how are we to discover whether they are working at the same level? If they are not, how are we to harmonise the products of their analysis so that they become coherent? Can a workbench help us here? Or do we have to wait for a more refined and complete methodology which is not yet available? None of these. It is the same problem that the strategy study faced, and the missing ingredient is the concept of time. Consideration of timing can focus our attention very effectively on the proper level of granularity that is required. The following rules express the central idea:

- o Any activity which at a particular level of abstraction is seen as independent in time from other activities can be considered as an independent task for the purposes of analysis at that level of abstraction.

- o At a higher level of abstraction (perhaps as seen by senior management) it is allowable to combine a group of related processes into a single process, making this process appear independent in time at this higher level of abstraction. At a lower level of abstraction it is possible to divide a process into a set of related processes which are each represented individually, provided that they are time-independent at this lower level of abstraction.

- o Any operation within a system which (at a certain level of abstraction) must be immediately preceded by or succeeded by a related operation has to be a part of a process, and can not be a process in its own right. Instead, it must be regarded as a step within the lowest level of truly independent processes above it.

The question of what we mean by "immediately preceded by or succeeded by" is an important one, and directly reflects the concept of real time. We mean that the operations are connected in time, on a timescale that is appropriate to the level of abstraction employed. For a warehouse operative the appropriate timescale is seconds and minutes, so that getting onto a fork lift truck and pressing the accelerator becomes a part of the same process. Receiving goods and recording their details do not necessarily have to be done in the same day, and they can be considered as separate processes.

The problem solved:

By employing this concept, and by supporting the analysis with data flow diagrams which were levelled in this way, the strategic study was able to proceed, and to conclude its analysis in an orderly way. The results were acceptable because they were coherent, and it was possible to adjust the level of abstraction to suit the person with whom one was speaking.

It is important not to confuse what was done with conventional structured systems analysis however, because it was not directed at the same aim. The technique was the same (top- down decomposition) and the tool was the same (the data flow diagram) but the problem that was solved was really quite different to the typical systems analysis problem.

It seems likely then that the strategic analysis of information systems will not necessarily be straightforward, and that a clear discipline will be required. The granularity of any strategic analysis will tend to fall below the sub- system level, and the assembly of the resulting system development tasks into sensible work packages for project teams will be a major challenge.

It will always be possible to compromise by implementing the whole of a sub-system using a single package, or by bespoking a complete solution where the effort of incorporating existing parts seems too great. And yet systems are becoming larger, and the penalties for not mixing and matching bespoke, packaged and existing systems will become far greater too.

Strategic analysis will require a much more flexible approach to system development. This will be reflected in a need to deal with the parts of systems differently, to acquire them from different sources, and in many cases to reuse parts of the system that today would be discarded.

Reuse of Software and Other System Components

Perhaps the the reuse of software and other system components will be the most important technique for development in achieving overall productivity. Avoiding doing twice what we need do only once is clearly a problem. Reuse is not a new idea, but there are many indications that it might at last become a viable option:

- o Methods for specifying the user requirement are much more rigorous and distinguish between the problem and the technical solution. A problem that is defined independently of the solution can be re-implemented in different contexts.

- o Likewise, technical specifications are more rigorous and are likely to be supported by workbenches. Specifications that are on workbenches are eminently reusable, given a degree of standardisation between workbenches.

- o 4GLs have not been universally successful, but they do provide a fast way of amending a system which almost matches the requirement, and they often separate the specification of function from the specification of data. Thus function and data become independently reusable.

- o The disciplines which allow us to manage the multiple use of system components and to relate them to their associated documentation and changes (particularly configuration management) are becoming better understood.

An enterprise that embarks upon reuse will have to address a number of different technical issues, including deciding what are the prerequisites for reusability in components and what methods and techniques will support reuse. Prerequisites for reusability in components:

There is sometimes a mistaken view that reuse is applicable only to program code. All of the components of a system are potentially reusable, including parts of the initial business requirements specification, system specifications, program specifications, data structure specifications, source code modules and test plans. It is well known that there is a hierarchy of objects which make up any system; there are a number of detailed models in the literature which attempt to describe these objects and their relationships, but no universal agreement as to their details. Consideration must be given to the level of abstraction in systems specifications, the degree of granularity in models of the objects in systems, and the precision and degree of formality in the specification process. The use of specific methodologies may facilitate or deny reuse.

The reusability of any particular component of a system will be affected by a number of characteristics:

- its level in the component hierarchy,
- its functionality,
- its external interfacing requirements,
- its environmental requirements, and
- the manner of its documentation or other description.

Methods and techniques for reuse:

Knowing what is required before components can be reused is one step, but only the first. In actually achieving reuse, there are several critically important matters to be dealt with.

Systems which are in part reusable, or which employ reusable components, may have to be represented in a special way in order that they are properly identifiable. The greatest single reason for difficulty in achieving reuse is the problem of identifying and recovering components. Although there has been some progress in the representation of systems using formal and semi-formal methods, it is not yet clear whether the available techniques are sufficient. Reusable components have to be conceived, specified and constructed. Creating reusable components may include significantly different tasks from those with which we are familiar.

The corporate view:

As well as the technical analysis of reuse, there must be some consideration of the non-technical issues. Reuse is not just a technical matter, it is important at the strategic levels of information system management. Reuse will be appropriate only in certain areas and not in others - it is most likely to be of benefit in the strategic and operational quadrants of our model.

Another non-technical issue is that of ownership where different parties are involved. It will be very important to both sides - the supplier organisations and user organisations - to understand who retains how much ownership of which parts of a system.

Open System Construction - The Need for a Universal Model

This discussion has encompassed many facets of system development. Starting with the turbulent times in which we live where information systems can be the deciding factor in the success of an enterprise, and by looking for areas of improved productivity, we have identified that strategic planning in information systems development is a pre-requisite if we are to avoid wastage, and that in each of the disciplines of development there are interesting ways of addressing the productivity issue. Further, a careful strategy for system development is going to make new demands upon our development staff because the optimal solutions are likely to require a mixture of bespoke, packaged and reused system components.

The sources of these components are unlikely to be all within our own control. The degree to which we depend upon third- party suppliers of software will increase, and there may also be an increasing need to co-operate with other computer users as the trend towards integrated systems spills over the corporate boundary into the outside world.

At the start of the paper a wish-list of desirable things was proposed:

- o a more productive way of implementing systems;
- o a proper strategy for sorting out and prioritising the backlog of systems development;
- o better ways of integrating new and old systems, and of avoiding doing our jobs twice over;
- o better ways of matching our requirements to the capability of packages.

If we are to realise these wishes we will have to recognise the need to standardise. In the final analysis it is the interfaces between co-operating partners that have to be agreed. We need to distinguish between what is done and what is produced. We must agree the activities within development and the nature of the deliverables that they are expected to produce. What is needed is a framework for system development which does not require or presume that people will conform at a detailed level, but which provides reference points around which partners in system development can discuss specific activities and deliverables within their own context. In this way the programmers will avoid doing things several times over, designers can take maximum advantage of existing system components, and analysts can specify requirements in a way that will not preclude any source of systems: bespoke, reused, or packaged.

In the ideal case, a proper framework would ease many problems:

- o Those responsible for procurement could avoid the tortuous business of defining from first principles what it is that is being procured.
- o Those offering products and services could identify them properly, and place them accurately relative to others.
- o Those overseeing development (project management, quality inspectors and accountants) could do their audits, calculations, tabulations and analyses using criteria and metrics which will render the results comparable from one project to another, and perhaps even from one organisation to another.
- o Those undertaking development could minimise the confusion that arises between the different functions and job titles (indeed job descriptions could be written unambiguously and optimally).
- o Those who train our new DP specialists could provide a balanced syllabus that recognises actual practice, retains flexibility in newly formed attitudes and fully supports the rapid rate of development currently to be seen in the technology of system development.

In effect we would open up the process of whereby we construct information systems, and make it much more fluid. There would be more options, and more points at which options are available. Mixing and matching system software components from different sources will benefit the user in the same way that mixing and matching hardware does.

At the spring 1986 conference on Software Engineering tools at Lancaster University the closing words included a heart-felt plea for standards. The ACARD report on software as a key to UK competitiveness devoted a whole appendix to the question of standards, and offered stability, mixed software supply and quality as primary targets for the standards effort. There are even new pressure groups forming. The IT Users Standards Association (ITUSA) has spawned the Users Standard Forum for IT (USFIT), which is gearing up to inject new effort and purpose into the standards making process. But standards themselves are not the total answer, because the standards themselves must be coherent and well conceived. We come back to the need for a framework.

There is a new international effort to develop the required framework for standards. The idea is to prepare a reference model. This will not itself be a standard, but it will enable the production of coherent and well-constructed standards in those areas that need them. It is comparable to the development of the OSI (Open System Interconnection) reference model for communications standards. It is intended that the reference model for software development will, if it is developed, properly reflect current practice and problems. It will provide a means to evaluate existing standards as well as the need for new standards - these new standards will therefore be purposeful, effective, and well co-ordinated.

The OSI model was for Open System Interconnection. The proposed new software development reference model is aimed at Open System Construction. This is a worthy ambition, surely. Time will tell us whether it is achievable.

References:

[Bytheway 1984]	Andy Bytheway (internal F International technical review paper)
[Collins 1987]	Tony Collins, "Revenue report taxes imagination", Computer News, 19th February 1987
[Gilb 1983]	Tom Gilb, "Increasing software productivity", Data Processing, Vol 25 no 7, September 1983
[McFarlan 1984]	F Warren McFarlan, "Information technology changes the way you compete", Harvard Business Review, May- June 1984
[Porter 1979]	Michael Porter, "How competitive forces shape strategy", Harvard Business Review, March-April 1979
[Sweet 1987]	Pat Sweet, "Pictures worth a thousand words" (concluding paragraphs), Computing, 19th February 1987
[Wyman 1985]	John Wyman, "Technological Myopia - the need to think strategically about technology", Sloan Management Review, Summer 1985
[Youll 1984]	David You