Information Coordination:

The Management of Information Models, Systems and Organizations

Richard Veryard

First published by Prentice Hall, 1994
ISBN 0-13-099243-7

Chapter 2: Concepts
Part b: Information Concepts

2.1 Introduction

This document is one of a series of extracts from my 1994 book on Information Coordination.

This document contains the second half of chapter 2. It applies the general concept of coordination (introduced in the first half) to the coordination of information systems and projects.

...

2.5 Purposes of Information coordination

The first half of this chapter has been discussing coordination in general. We can now apply this to information coordination in particular.

The potential benefits include: Information Sharing, Reuse and Comparability.

These benefits may be required to make the business more efficient (by making the information systems more efficient or effective). This should make the benefits quantifiable, and directly comparable with the costs of achieving them. Sometimes the business can benefit from the economies of scale available from Information Processing, or economies of scope.

Alternatively, they may be required to fulfil some aspect of the business strategy directly.

Measures to achieve strategic benefits tend to be more difficult to cost-justify than measures to increase operational efficiency, and should be planned as part of an Information Systems Strategy Planning exercise.

2.5.1 Levels of coordination

Coordination between information systems is sought for two reasons. We shall refer to it as **enterprise coordination** if the lack of coordination would affect the business directly, and **technical coordination** if the lack of coordination would merely increase the costs/risks or reduce the quality/productivity of computer systems development and/or operation.

Strategic business benefit comes from enterprise coordination. For example, lack of enterprise coordination may increase costs of specific business processes. Some business opportunities may be not worth doing at all in the absence of adequate coordination, because costs are too high.

Copyright © 1994 Richard Veryard

Technical coordination may deliver either enterprise coordination, or system lifecycle cost savings. Enterprise coordination will often (but not always) require technical coordination.

Thus it is important to understand the difference between enterprise coordination and technical coordination. Since this difference is often ignored by technically minded information engineers, I shall devote one section to examples of enterprise coordination, and the following section to examples of technical coordination, before going on to discuss how both may be achieved.

2.5.2 Enterprise coordination

There are several types of enterprise coordination required by a typical business enterprise, affecting key business performance measures such as market share and profit.

Often, it is part of the business strategy to establish and maintain such coordination. This provides a powerful argument to information systems planners, to ensure that the computer systems support the required enterprise coordination.

Single point of management

Objects of interest to the business are managed in a consistent way, by concentrating all decisions about the object in one place. For example:

i. Customers have a single contact point, allowing all business conducted with that customer to be coordinated and optimized.

ii. Products have a single management point, allowing all development, manufacturing and marketing of a single product to be carried out in a coordinated way.

iii. Functions are coordinated - e.g. marketing campaigns for different products may be coordinated - e.g. funds management may be coordinated, to minimize the firm's exposure to currency fluctuations

iv. Processes are coordinated - thus ensuring that business processes are carried out effectively and efficiently - e.g. shipments are combined to reduce total delivery mileage - e.g. inspection is carried out before assembly, to reduce wastage

Some of these decisions may be centralized at head office; others may be allocated to branch offices, but in a clearly defined way.

Force business to coordinate

Two systems development projects need to talk to each other, because the business areas themselves need to talk to each other. Sometimes there is a lack of communication within the organization, and the information systems are being developed to alleviate this. We saw some examples of this in Chapter 1.

Copyright © 1994 Richard Veryard

In many organizations, there is a push towards devolution and empowerment of staff. Getting decisions made at the point of action involves coordination of various concepts to support the decision. These concepts were perhaps previously understood by different people in different departments. Business Process Reengineering may involve breaking down these private domains.

Consistency of behaviour

If a company is selling the same product in several different countries, it may attempt to set different prices in each country. In other words, the pricing is a local decision.

However, if there are significant variations in price from one country to another, multinational customers will purchase wherever the product is cheapest, and then reship. Traders may even buy in the cheapest country, and sell in the more expensive countries, thus undercutting the company itself. Some computer manufacturers attempt to control these 'grey imports' by retaining a monopoly on repairs; they refuse to provide spare parts for computers except in the original country of sale. Other manufacturers merely ignore the problem. For some companies, however, it can be a serious drain on profits.

Wider examples can be found involving inter-organizational coordination (e.g. price-fixing cartels), or even inter-governmental coordination (e.g. harmonization of taxes and duties).

Consistency of data

For an organization to communicate effectively, it is necessary for them to speak a common language.

When data are summarized and compared, it is useful for them to have been calculated in a consistent manner.

A long time ago, a computer manufacturer designed a new range of tape drives. These were to be linked in a chain, with a vacuum pump at the end of the chain. The tape drive had a part number (say 4711), and the vacuum pump had a different part number (say 5822).

When sales and marketing started to sell these tape drives, however, it was decided that the vacuum pump could not be separately priced. The price list included a tape drive without vacuum pump (part number 4711) and a tape drive with vacuum pump (part number 5822). Salesmen were instructed that each order should include at least one tape drive with vacuum pump. Orders were taken and passed, in the usual way, to the production section for manufacture and shipment.

Customers who had ordered two off part number 4711 and one off part number 5822 expected to receive three tape drives and a pump. However, the production and shipment systems expected the customers to be satisfied with two tape drives and a pump. When the first customer complained, this was attributed to clerical error. By the time the real cause of the problem had been identified, it was already causing serious embarrassment to the company.

Copyright © 1994 Richard Veryard

Comparability of business

Consistent measurement and presentation of business performance means that the performance measures from different parts of the business can be fairly combined, compared and contrasted. This benefit relates to management's ability to exercise strategic and tactical control over the business.

2.5.3 Technical coordination

The Roman architect Vitruvius, a contemporary of Jesus Christ, identified three elements of quality: **firmness**, **commodity** and **delight**. I was surprised (and delighted) to find this definition of quality being repeated by Bill Gates, founder of Microsoft[1]. For software, Gates glosses the three components as follows:

Firmness	***Consistency***
Commodity	***Be worthy of the user's time and effort in understanding it***
Delight	***Engagement, fun***

Box 1 From Vitruvius to Gates - definitions of quality

We can see enterprise coordination as connected with the benefits to the enterprise, the fit-for-purposeness of an application system, which Vitruvius calls its commodity. Firmness/consistency includes a number of technical quality characteristics, including reliability and maintainability. This is closely connected with what we are here calling technical coordination.

The following benefits are related to the operational efficiency and effectiveness of the information systems themselves.

Data quality and integrity

Consistency of data can be important, even in situations where it does not bring any direct business benefit. A standard data format may simplify data processing, may simplify aggregation and comparison and analysis. And it allows data and system components to be reused.

Reuse of data

Information Sharing means that the same information becomes available to different parts of the business. Since the acquisition and maintenance of information is itself an expense, it benefits the organization if the same information can be exploited many times. Once data have been entered into a computer system, it is a waste of time and effort to enter it into another computer system. Furthermore, when several different parts of the business are using the same information, errors and inaccuracies will be more quickly noticed and eliminated.

Copyright © 1994 Richard Veryard

In some situations, lack of integration between systems results not merely in inefficiency but in irreversible loss of data.

System	*Data lost*	*Reason*	*Consequence*
Credit card system	Date of purchase	A customer buys some goods from a merchant, using a credit card. The customer's credit card slip is transmitted electronically from the merchant to the bank. There is a minimal interface between the merchant and the bank, designed many years ago, which only includes the amount and omits the purchase date.	The customer bill doesn't show the date of purchase, but the computer entry date. This causes the customer some confusion, and forcing the customer to record or reconstruct the actual date.
Job costing system	Product data	Job costs are calculated on the basis of timesheets completed by engineers. These engineers are reluctant to spend extra time completing timesheets, and there is no interface between the scheduling system (which does maintain the relationship between jobs and products) and the costing system.	This means the job costs have to be apportioned between products on a semi-arbitrary basis.

Box 2 Examples of lost data

Reusability of system components

Reusability means that Information System components, once built, can be used many times. This includes technical components (such as programs and algorithms, data definitions and routines) and also human components (such as user skills). As with information sharing, reuse enhances quality (but only if the reused components are appropriate and properly used).

Development costs can be saved by having standard components. It reduces the number of design decisions, and reduces the amount of testing (since the test effort can be reused).

Conservative fashion (also known as tradition) uses standard solutions rather than solve every problem from scratch. An often arbitrary decision may be made, which everyone then adheres to. Standards often have great value - they offer stability, but at the risk of inertia.

Some systems are built with high levels of reuse, which implies low levels of redundant functionality. This is thought to make the systems cheaper to maintain and enhance, since changes will only have to be made in one place, rather than in several places.

Copyright © 1994 Richard Veryard

However, if we see systems (of human-computer activity) as being parts of learning organizations, and if we take the ideas of General Systems Theory seriously, we should consider the principles of redundancy of function and requisite variety[2]. "Redundancy (variety) should always be built into a system where it is directly needed, rather than at a distance."[3]

Economies of scale

One of the reasons for sharing resources across several systems (or several organization units) is to gain economies of scale. For example, since the organization incurs a cost for each DBMS used (both software licences and internal support costs), there may well be a cost advantage if all organization units use the same DBMS.

However, if the different organization units have different requirements, it will sometimes be the case that the cost of compromise (i.e. using a DBMS that is inefficient for the given purpose) is greater than the cost savings from the economies of scale. These are often extremely difficult calculations to balance.

Economies of scale in decision-making

If you are going to buy a machine, and the choice is between spending $1000 on a cheaper model, or $1500 on a more expensive model, then you would want to consider whether the additional power and functionality of the more expensive model was worth the extra money. Suppose that the only way you would know for certain is by carrying out a benchmark study. If the study itself will cost more than $500, then it would be more sensible to do without the study and simply buy the more expensive model. (Many companies are prevented from doing the sensible thing, because capital expenditure comes from a different budget than evaluation studies, but that's a different problem.)

However, if you want to buy a thousand machines, it could be worth spending $10,000 or more comparing the options, gathering the information to support a better decision. Thus we can get economies of scale in decision-making, by sharing the information costs between a thousand separate purchasing decisions. This will only work, however, if all the thousand purchasing decisions take proper heed of the supporting information. Thus, to get these economies of scale, we seem to need standardization.

Uniformity of control

Uniformity of functionality and controls makes for greater user-friendliness.

Let us consider some analogies. When you drive a strange car, there is uncertainty about the various levers on the steering column. Sometimes you want to signal a left turn, and operate the windscreen wipers instead. Or in a strange hotel room, you cannot find the light switch in the middle of the night. Faced with such inconveniences, many people would prefer that all cars had the same controls in the same places, or that all hotel rooms were laid out in the same way.

Or consider differences in the highway code for car drivers. In some countries (including the UK), when joining a roundabout, you give way to people already on. In other countries (including France), the people on the roundabout give way to those wishing to join (priorité a droite). Some countries don't have roundabouts at all. Some counties don't even have a highway code.

Copyright © 1994 Richard Veryard

An organization may desire uniformity for its internal operations. This enables staff to be interchanged ad lib between different functions, and simplifies the development and maintenance of computer software. But this argument may not apply to the external information systems, the so-called strategic ones. Standardization may make it easier to poach market share from your competitors, but also makes it easier for your competitors to poach from you. So standards are two-edged.

2.6 Methods of Information coordination

2.6.1 Achieving enterprise coordination in an organization

Enterprise coordination requires clear and appropriate coordination policies, determined (usually) as part of the strategic planning process, and implemented (often) through layers of management. We shall return to planning issues in Chapter 4, and to the political issues in Chapter 8.

2.6.2 Should technical coordination match enterprise coordination?

We have said that the purpose of technical coordination is often to achieve some specific enterprise coordination. We have also seen that there are three main coordination mechanisms: hierarchical, market and network.

This raises the question whether the technical coordination should take the same form as the enterprise coordination. Does it make sense, for example, to implement a network of information systems to serve a strict hierarchy, or a hierarchy of information systems to serve a market?

Sometimes not only enterprise requirements but also technological factors may influence the architecture of the information systems. A technical architecture that matches the enterprise architecture may simply not be available or feasible.

However, experience suggests that although implementation of mismatching technical architectures is possible, it will be much more difficult to manage. The reasons for such a technological mismatch need to be well understood, and carefully explained to the stakeholders, since they will be required to alter the way they plan and control the information system support. Hierarchical systems, for example, have to be planned and designed hierarchically, and if no hierarchical decision-making and issue-resolution mechanism exists, it may have to be created specially for the purpose. We will return to this question in the final chapter.

2.6.3 Data ownership

In a hierarchy, the data are owned at the top. Agents are merely the custodians of data.

Copyright © 1994 Richard Veryard

In a market, the data are owned privately by the agents, who may make formal contracts to provide them to other agents.

In a network, the data are owned privately by the agents, or collectively by the local network. Communication within a local network may be informal; communication outside the local network may be based on wider network transactions or market transactions.

Most organizations combine the hierarchical principle with the market principle. The concept of internal cross-charging is common to most commercial organizations, which implies market transactions between parts of the same organization.

This tells against the oft-cited principle of treating data as a corporate resource. If a cost-centre or profit centre has expended effort in collecting and processing some information, why should it provide this free to other cost centres? Since other services are subject to cross-charge, why should information services be exempt?

2.6.4 Coordination within / between systems

Sometimes, coordination can be achieved by integration and homogeneity. Sometimes it is more appropriate to tackle the symptoms of contradiction, rather than to eliminate them by full integration. There are several ways of doing this, including insulation, redundancy, and standard interface protocols - “open systems”.

Homogeneity

Until recently, the majority of large application systems were designed and implemented as centralized host systems. Such systems can generally be assumed to be **homogeneous**. The characteristics of homogeneity can be seen in Box 3:

Copyright © 1994 Richard Veryard

- ***Single global name space***
- ***Global shared memory***
- ***Global consistency***
- ***Sequential execution***
- ***Failure is total***
- ***Synchronous interaction***
- ***Locality of interaction***
- ***Fixed location of components***
- ***Direct binding***
- ***Homogeneous environment***

Box 3 Assumptions of homogeneous systems

Insulation

Insulation means that parts are shielded from other parts, keeping their interactions to the minimum necessary. In the design of houses and offices, we can see several examples of this:

- Sound insulation enables people in adjacent rooms to sleep at different times, or to listen to different music. (People in open-plan houses have to coordinate their lives much more.)

- Firewalls or firedoors to contain and compartmentalize the risk of failure.

Analogous insulation possibilities exist in information systems. For example, databases are often designed to contain 'firewalls'. This is an important technique, and we shall return in later chapters to the techniques for placing these firewalls, and for providing the necessary links across them.

This brings us back to the difference between a market and a network, namely the degree of formality in the interface. In a network, there is 'trust' across the interface. For example, the data quality within one application system may depend on the data quality in another application. Problems of data integrity can easily ripple across the network. If there is a serious software crash, it may be difficult to trace the source of the fault.

Copyright © 1994 Richard Veryard

A market interface, on the other hand, is much less permeable to data quality problems. As in a real market, applications must apply the principle of 'buyer beware', revalidating data integrity rather than relying on the quality of the data received. Software faults can be much more easily located.

A market interface may also be less permeable to other problems, such as software virus infection.

However, such insulation has its own costs, since much of the validation will arguably be redundant. The interface can be regarded as a formal contract or agreement, and the double-checking of data quality can be regarded as the cost of enforcing the agreement.

Redundancy of function

In some circumstances, change is impossible without redundancy. Many real-time computer systems are designed for constant operation: you just cannot switch off the system that controls a nuclear power station, even for a few seconds; such systems always incorporate the principle of redundancy, usually by duplicating parts.

Furthermore, self-organizing or even self-improving systems must include some redundancy. Whereas redundancy of parts is a feature of mechanistic hierarchical control systems, in which spare parts are added to a system, **redundancy of function** is a feature of intelligent networks in which extra functions are added to each of the operating parts, so that each part is able to engage in a range of functions. Requisite variety is built into the system where it is needed, rather than at a distance[4].

Standard interfaces

The concept of **open systems** has attracted much interest from purchasers of computer hardware and software, and considerable hype from suppliers. The vision is that if there is a standard interface to which all suppliers conform, it will be possible to plug anything to anything else. All hardware and software becomes interchangeable commodities.

The IT industry has discovered that defining and agreeing such a standard interface is easier said than done. It is possible (but unnecessary) to be extremely pessimistic or even cynical about the open systems movement. Suppliers have an uneasy dilemma: conform exactly to standard, provide no non-standard features (even if they are better than standard), and compete solely on cost, performance and delivery. Or try to provide something extra, and be accused of rebelling against the open systems community.

Analogies abound. There is no worldwide standard voltage for electrical appliances, and there are several different standards for sockets. It would of course be dangerous to standardize the sockets without first standardizing the voltages. But we probably have to accept the fact that the different local standards are now too entrenched, for a worldwide standard ever to be implemented.

Copyright © 1994 Richard Veryard

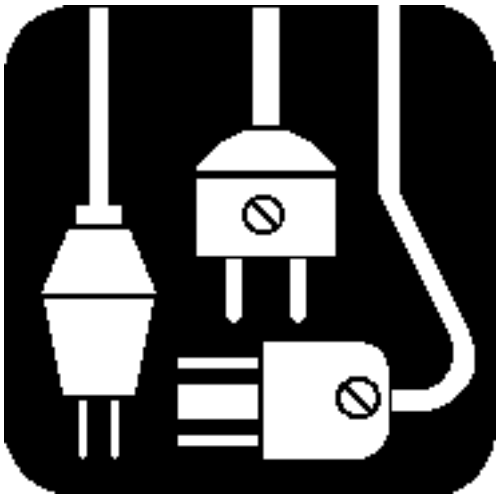

Figure 1 No world-wide standard for electric plugs

Standards may be restrictive. Designers have been attempting for 50 years to replace the QWERTY keyboard without success.

The railways provide another interesting example. By a feat of network coordination, most of Europe operates on the same railway gauge. However, nineteenth century Russia decided to adopt a different gauge, for fear of invasion from the West. Standardization enables the strong to attack the weak, but it also allows the small to infiltrate the large. Standardization is therefore a highly political issue, although technocrats refuse to recognize this.

In IT, standards have been developed for Electronic Data Interchange (EDI), which provides a fairly limited language for communicating data between organizations. This is a useful basis for inter-company coordination, but does not provide the coordination itself.

For example, a small software company Alpha may subcontract the duplication of floppy disks to company Beta, the printing of manuals to Gamma, and the warehousing and shipment to Delta. There must be communication between the companies, which could be paper or electronic. It would be fairly straightforward to design a logistics system for Alpha, which sent orders to Beta, Gamma and Delta, and received status reports back. The logistics system would then be linked to a payments system, that would receive invoices from Beta, Gamma and Delta, and initiate cash transfers back. The idea of EDI is to make such interfaces easier to build, because there are common protocols for such business entities as orders and invoices.

Copyright © 1994 Richard Veryard

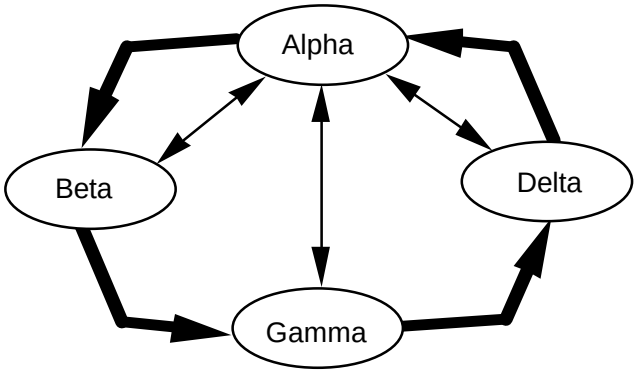

Figure 2 Single company Alpha being serviced by Beta, Gamma and Delta
[Fat lines show material, thin lines show information]

But suppose there are several software companies, Alpha-1, Alpha-2, etc, using the same trio of Beta, Gamma and Delta. If Beta, Gamma and Delta liaise directly, they may be able to achieve significant economies of scale. For example, duplicated disks and printed manuals could be delivered to the warehouse in larger batches. An Alpha-controlled logistics system would not support this; a network of symbiotic logistics systems would be needed. This would be much more difficult to design and implement, and goes way beyond simple EDI.

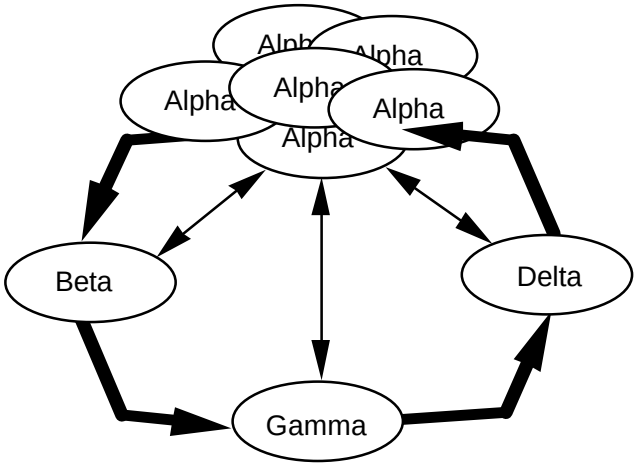

Figure 3 Many companies being serviced by Beta, Gamma and Delta
[Fat lines show material, thin lines show information]

Near coordination

Earlier in the chapter, we discussed the concept of near coordination, using the example of the overhead walkways in the City of London.

How is near coordination possible with information systems? Interfaces can be imperfect or incomplete, requiring manual intervention. Interfaces may involve a delay or cost, instead of allowing cost-free real-time update. With such interfaces, there is near coordination, reducing but not entirely eliminating the symptoms of contradiction.

Copyright © 1994 Richard Veryard

Open Distributed Processing

Probably the most influential work in the area of open distributed processing has been done under the umbrella of the Advanced Network Systems Architecture (ANSA). The ANSA model adopts several principles for market-linked systems[5], based on the client-server model of computing. These principles characterize the interfaces themselves, as well as the mechanisms for implementing and controlling them.

The ANSA model has also been influential in introducing the concepts of trading and federation[6]. In **federation**, separate systems can interoperate but still maintain control of their own domains. **Trading** refers to the process by which clients can use the attributes of a service as well as its name to find the appropriate servers and services on a network. A **trader** (or **broker**) acts as an intermediary between client and server. These concepts belong without doubt to the market mode of coordination.

Copyright © 1994 Richard Veryard

SEPARATION

- ***Assume every module is on a separate physical platform (the most complex case for the designer). Optimize by sharing platforms (co-location), but without changing the internal design of any module.***
- ***Each system is responsible for processing its encapsulated data***
- ***All interactions with services are through defined interfaces***
- ***All detected interaction faults and failures are named and reported***

HETEROGENEITY

- ***Assume heterogeneity***
- ***Abstract away from unnecessary diversity, while still retaining the benefit of specialization***

FEDERATION

- ***Each system controls its own policies and services locally***
- ***Cooperating systems negotiate the sharing of services***
- ***Cooperating systems identify the available services via a context-relative naming scheme***
- ***A trading facility is provided through which federated cooperating systems can organize and control the sharing of services***

Box 4 ANSA principles of distributed systems

These principles were behind the NASA Astrophysics Data System (ADS), which is already claimed to be the largest distributed system in the world, and is expected to incorporate over 100,000 users within 2 years. It uses a wide variety of hardware and software, to allow scientists all over the world to share text, numerical data and images from outer space.

Copyright © 1994 Richard Veryard

The problem faced by scientists before ADS was the fragmentation of data and processing, residing in different formats on a wide variety of machines. It was difficult to discover what data there were, or what applications had already been written by other scientists at other institutions.

For the end-user, ADS provides 'one-stop-shopping' - the ability to retrieve and process data regardless of its location or technical format. For the computer industry, ADS demonstrates the feasibility of such networks with heterogeneous hardware and operating systems, and the feasibility of evolving such networks without disrupting existing operations using existing standards and protocols.

2.6.5 Coordination within / between projects

If two or more parallel projects are building or maintaining systems that will need to coordinate, it is almost certain that some coordination will be needed between the projects.

Since projects are human organizations, they may also be coordinated as hierarchies, markets or networks. We might expect that hierarchically coordinated projects are best able to build hierarchically coordinated systems, and so on. Of course, it isn't quite as simple as that. We shall return to this issue in Chapter 5.

2.7 Summary of Chapter

Now we know what coordination is, why (and how much) we want it, and (broadly) how it can be achieved.

Now we are ready to look at the concepts and techniques of information coordination, creating and managing the underlying information models for:

- scoping and coordination of projects
- scoping and coordination of systems

The next two chapters will look at planning and scoping of projects and systems, and therefore at the scoping of information models.

Notes

[1] Tom Peters, *Liberation Management: Necessary Disorganization for the Nanosecond Nineties* (New York: Alfred Knopf, 1992) p 688

[2] G. Morgan, *Images of Organization* (London: Sage, 1986) p 98 ff

Copyright © 1994 Richard Veryard

[3] G. Morgan, *Images of Organization* (London: Sage, 1986) p 100

[4] G. Morgan, *Images of Organization* (London: Sage, 1986) pp 98 ff, citing Fred Emery and W. Ross Ashby

[5] ANSA Reference Manual (Cambridge UK: Architecture Projects Management Ltd, 1989). See also D.S. Marshak "ANSA: A model for distributed computing" (Network Monitor, Vol 6 No 11, November 1991)

[6] ANSA Reference Manual (Cambridge UK: Architecture Projects Management Ltd, 1989)

Copyright © 1994 Richard Veryard