# Architectural Considerations in an Enterprise Georelational Environment

## By Milton Lofberg

**Principal Consultant**

**SPATIAL*info* Pty Limited**

*Presented at the annual GITA Australia-New Zealand Conference, Aug 1999*

## Abstract

AM/FM/GIS systems have traditionally been oriented towards engineering, or the capture and management of 'as built' records. These systems have been the domain of professionals who exercise their network skills within a microcosm of the business IT infrastructure. Implementation of these systems within the Enterprise has been characterised by purpose-built, relatively inflexible or labour-intensive "glue" between disparate systems.

Organisations are increasingly required to respond flexibly to changing business requirements. Today, the Enterprise IT infrastructure is more and more able to meet the demands that business presents, providing greater flexibility and nimble response within shortening timeframes. What of AM/FM/GIS systems? These too must be integrated with other business systems, often presenting the challenge of leveraging industry standards in the areas of IT skills, relational data modelling, protocols and software building blocks such as RDBMS.

This paper examines this changing environment, implementation trade-offs, and the continuing importance of the "glue" when integrating business systems, together with the notion of distributed systems, users and information.

## Introduction

Technology Architecture is about thought, collection and synthesis of information, design and trade-offs. To come up with a specific Technology Architecture – for *your* organisation – you need to be technically aware. However, whilst this awareness is required, you need not be a 'systems guru' to understand the business needs and drivers behind the concept and plan to achieve a result. You need 'Big Picture Thinking'. This is a discipline that encapsulates well-known subjects such as General Systems Theory (the original GST!), which existed well before the computer era[1].

There is no such thing as 'Architecture without Tears'. Managing a Technology Architecture involves a long term commitment to ongoing team work, thinking, analysing and defining a framework for thinking about technology that will meet the needs of the organisation – long term. So instead of focusing on piecemeal short-term expedients ("I'll just buy what everyone else has") or 'Big Bang', everything at once changeovers to new technology, your organisation can implement systems intelligently, and incrementally. You will increasingly leverage data investment in proactive, perhaps simple and diverse ways, retain corporate information in a corporate repository (and I don't necessarily mean a piece of technology here) and last but not least, promote the use of the right skills within your organisation.

Having a Technology Architecture provides benefits for implementation of systems across the enterprise. The development of an Enterprise System (that is, a physical, operational technology infrastructure) needs an approach that is fundamentally about working from the 'whole to the part'. In other words, you can work on the 'parts' if you understand their role as part of the 'whole' – like a jigsaw puzzle. Alternatively, if you aim at nothing, you can be sure that you will achieve that.

[1] General Systems Theory was developed by biologist Ludwig von Bertalanffy in the 1940's. The theory conceptualised behaviours of, and relationships between, entities as systems having interacting components. A Big Picture could be formulated and documented to describe any such system. A 1956 book by Ross Ashby entitled 'Introduction to Cybernetics' arguably took GST into the modern computer world.

Another concept that can often be given little more than lip service is that *the Business should always drive the Technology*. Greater business flexibility and nimble response is expected today as Utility organisations deal in an increasingly competitive yet regulated market. The landscape is changing within shortening time frames and increasing regulatory compliance, and competitive edge and responsiveness is likely to be short-lived (at best) or intangible (at worst). If my organisation has to be nimble, if the Business Processes within my organisation change, or Regulatory Compliance tightens, I have to be able to implement change in the Technology Infrastructure supporting my Enterprise. The issue then is 'how does my Technology Architecture support the evolution of the Business?' 'Am I enabled to innovate and be flexible and responsive to meet the needs of an increasingly competitive but regulated environment?' Technology must therefore be complementary to Business, never an end in its own right.

This consideration will also drive into other areas of thought and planning, potentially requiring some difficult decisions:

- What Business Systems do we actually have?
- What value do those Business Systems add?
- Are those Business Systems doing what we need?
- Is that Business System really Core Business?
- How do I rationalise my Business Systems?
- Do I really need that much sophistication in the system integration?

Today, Business drivers *are* being compared with the IT environment. Methods of ensuring Customer loyalty, or improving Customer Service or Operational Efficiency or Decision Support are occupying significant mind share. Information Technology Departments are under pressure to deliver more meaningful business outcomes from the IT infrastructure, and often react with dismay at the integration pressure brought by changing corporate and user base requirements and expectations.

For example, Executive Management is asking how Corporate Information – including spatial information – can be made available on an Intranet. The Corporate Group needs access to compliance reporting data. The Works Management Group is upgrading their FMMS™ system. The Finance Department is implementing Oracle® Financials™. There is a new SCADA system in the wind. The Call Centre wants "access to maps" (sic). Integration of Business Systems in this climate can become daunting. To add to the fear wrought by prospective change, some IT Departments are outsourced, in the hope that greater responsiveness, innovation and efficiency, if not economies, are to be had.

# Technology Architecture

As you can already see, where Technology Architecture is concerned, it would be useful if we could cease thinking purely in terms of 'technology', or 'band-aid' solutions (even though these might help us out considerably, yet temporarily in some cases). To fuel this change in mindset, I'll now abbreviate the term "Technology Architecture" to merely "Architecture".

Now that you can see where I am coming from, I'll draw a parallel. Let's also modify our thinking when considering geospatial systems. Let's forget the once true, but oft-abused mindset that AM/FM/GIS systems are different. The concept of AM/FM/GIS utilising architectures that are common in the broader IT industry has been viewed as all but impossible for most of the last decade, despite available technology. Let's forget that ideology and move into the 90's-proper and the millennium ahead.

It may be helpful to think of Enterprise Architecture in terms of your 'Dream Home'. You wouldn't attempt to construct a house without plans. The plans need to be driven by a 'concept' or 'framework', and a logical view of what you are setting out to achieve. You'll have an Architect assess the situation, sketch some options and then put some clearer definition on the proposal.

Now reality sets in. You or the Architect of your dream home might have creative ideals that don't fit your budget. Practical compromises are made. But you know that you want four bedrooms, lockup garage and a set of coordinated materials (bricks, tiles). You may choose to finish the interior of the rumpus at a later date. You will allow for a possible extension on the northern end of the house by sensibly siting the house on the parcel. Access to the back yard for the (future) pool is planned.

In terms of the skills required for the overall project, you're not an Architect or a Builder, but you understand the issues. You employed *them* to understand the details. Anecdotal or 3rd party evidence leads you to ask questions and make yourself aware of the risks. You juggle the aspects of your Dream Home. You know the Big Picture and your priorities, and your likely starting budget.

On this basis, your house proceeds pretty much as planned. Of course, in this situation, O'Reilly's Law often comes into play! So contingency or risk mitigation plans are also crucial.

Formulating an Architecture for an organisation is similar in some ways, although like all analogies is deficient in some ways. As noted earlier, Architecture is often viewed as a set of *technology* issues, whereas in fact the issues may also include a large component of Business Processes, internal politics, funding limitations, ageing systems, or even business mode (is your organisation the aggressor or the target in a takeover bid?).

It's worth noting that Architectures work at both the 'micro' and 'macro' level – if you have your thinking right. For example, an individual Business System will have an Architecture. The applications within that Business System will have their own Architectures. Aggregating these Business Systems will provide your Enterprise Architecture. The quest is to strive to achieve this kind of elegance, long term.

# Assessing Current Systems

Against this changing background, your organisation's current Business Systems will likely present many challenges. Despite having implemented many enabling technologies, there are many potential issues that you probably still face today:

- Data is in many databases. (If you are lucky, the physical databases are from a single vendor, or are each based on relational technology.)
- Data is probably duplicated and based on differing schemas.
- Data is keyed differently, or there is common key between logically associated databases.
- Interfaces between databases exhibit various states of sophistication.
- The network is perhaps old or obsolete.
- Network protocols are many and might include IPX, TCP/IP, DECnet, SNA etc.
- Many PC Workstations are unable to run the latest geospatial (or indeed office automation) software.
- Servers probably include several flavours of UNIX, OpenVMS and at least one AS/400 system.
- Many applications are classed as "legacy" (a loaded term) or require specific niche skills to maintain.
- The Corporate Intranet is a dream.
- The AM/FM/GIS data is either CAD data structured as Maps, or is embedded within data storage native to the AM/FM/GIS system (even if this storage is relational).
- Business goals have changed. The IT Department cannot currently meet the demands for change.
- Executive understanding and support for your geospatial systems has been erratic.

In other words, what poses as an Architecture for many existing systems is in fact a collection of loosely-knit Business Systems – factions of warring hardware, software, network components and staff. Let's be realistic. Technology acquisition hasn't always worked according to any sort of sound Big Picture thinking.

The early to mid '90s saw a focus on GIS systems as 'portals' to all corporate information. As a result, many organisations spent a fortune on an integration environment where their AM/FM/GIS applications are connected (hard-wired) to some other Business Systems but don't truly share a technology infrastructure at all. In some cases larger Utility organisations now find themselves the proud owners of purpose-built, difficult to maintain interfacing software applications, developed using esoteric, proprietary development environments, much of which can now be considered as 'legacy'. Over time, more and more interaction between systems means that these interfaces are liabilities – as support costs escalate.

Note that interfaces will always be required. The issue is whether or not the interface is easy or hard, smart or dumb, hard-wired or configurable. All of these contrasts will severely impact upon an organisation's ability to respond to changing business requirements. Skills are at a premium, and it makes sense to write software only where necessary, and above all, use commonly available skills and tools.

# Thinking about Future Systems

Assessing the state of your technology implementation could yield depressing results. However at this point the task is to develop a Big Picture. When you understand both the state of your organisation's technology, together with what has been implemented, some common threads should emerge, representing issues for resolution:

- Enterprise information access and information management – the establishment of one or more Intranets may be attractive.
- Management of heterogeneous systems – dealing with issues without merely reacting to change.
- Integration of Business Systems in a coherent and thought out manner – based on the right "glue".

In this situation, "Corporate GIS Access" projects are commonly promoted with varying degrees of success. This is where the Web Technology developments of the last 5 years start to provide huge potential.

As stated previously, I am suggesting that our thinking is flawed without the help of a Big Picture that has taken time and effort to develop. You need to take the state of the nation and examine it closely.

Looking across the Enterprise you will be able to identify Business Systems, together with where the interfaces between systems exist – or need to exist.

Each Business system has its own Domain, with its own users, data and technology. Therefore, it is important to remember that you only need to delve deeply enough into each system to achieve a required level of integration. You don't have to know every system's internal operation. All you need is the equivalent of an interface – ask "what can it give me, and how do I get it?" For example, were I to obtain the Asset Management software relational database schema, I only need to understand sufficient portions of the assets data model to achieve back-end (database) integration, perhaps using foreign keys or linking relational tables). The rest of the system can be viewed as a black-box, managed by its Custodian (elsewhere this paper discusses the notion of 'Custodians' – I may only need to ask the Custodian for help!).

We know that each Business System will have implemented its own database. We now want to establish an Architecture with the aim of ensuring that each Business System will not require hard-wired interfaces to other Business Systems. In other words I am suggesting that your Architecture must provide the information, technology and skills – to enable applications to work together for the Business with minimum fuss and maximum facility.

This is why the Big Picture thought process is so valuable. Business System integration can be independent of any single Business System. Integration by Architecture provides structure and reduces risk and cost. Any complexity becomes merely a solution of smaller pieces of the structure or limited to the interface between systems.

Fundamental to the Architecture is technology choice. The fundamental technology enablers in the Architecture provide the means by which Integration by Architecture is achieved. For example, it's no good assuming that 'back-end' or database integration can work for your Business Systems if those systems are not relational.

Establishing a new Architecture is a planned approach. You have to formulate it, then implement it. Later in this paper I'll cover some considerations for the implementation. I am also suggesting throughout this paper that we do not implement the new Architecture in one 'big bang'. Technology implementation under this scenario should be progressive, building on previous successes *and* include failures as lessons that provide valuable feedback into the expectations or components of the Architecture.

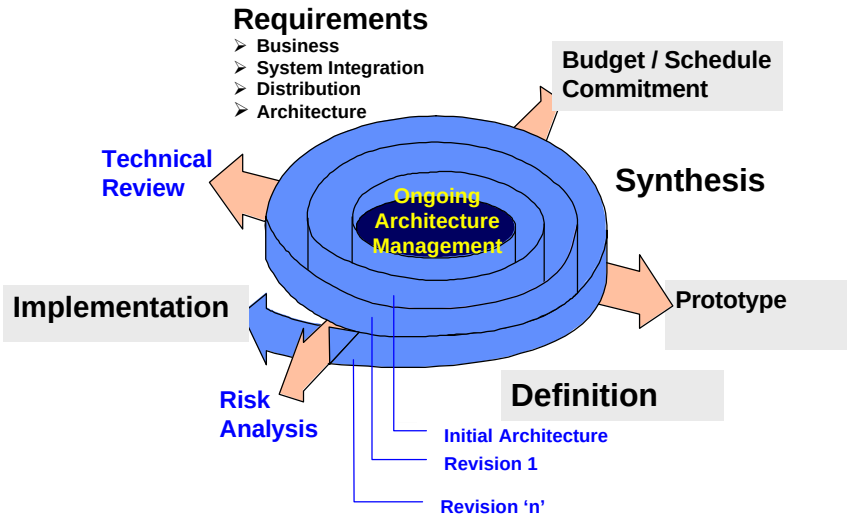


**Figure 1 – Architectural Evolution**

As a competitive Business is the main goal, the Architecture implementation will require some level of integration of information and applications. This means that existing systems must be accommodated in some way until such time as they can be migrated to the new architectural framework.

Some of existing Business Systems will work very well – if it's not broken, don't fix it. However, even if that system works well, it will still need to be integrated into the whole as necessary.

# Architectural Requirements

As I've suggested earlier in this paper, the success or failure of a Business System will be the by-product of the Architecture.

Another way of looking at the effect of Architectures is that you can't really critique them as wrong or right. It is merely that getting an Architecture "wrong" within the context of your organisation will negatively impact a Business Systems implementation or integration in some way. Your own business context is very important. For example, if you specify an Architecture that does not include the use of an open API, but there is frequent need to interface applications programmatically, you may have to commission expensive specialist contractors to write specialist code to interface to a particular application. If you had specified a minimum of database connectivity, the problem may not have existed.

My point is that Architecture significantly affects the subsequent phases of system integration, so the proper effort 'up front' pays long-term dividends. By thinking and formulating carefully, you have a much greater chance of mitigating implementation risk as a by-product of the Architecture.

I'll now spend some time looking at inputs to the Architecture.

## User Requirements

The Users of AM/FM/GIS systems are not generally concerned about Architecture. They will typically want to do their job. They will often express loyalty to their chosen application for personal reasons – skills developed, professional interest, fear of change, job security etc. The Architect will know that a 3-tier application is a better choice that a 2-tier application provided reconfigurable, reusable business rules are available from an AM/FM/GIS system. Again, context is all-important.

A corollary then is that your organisational context and the context of a Business System within the Architecture determine the trade-offs that you will make. In other words, it's a needs-and-context assessment when formulating and evaluating Architecture. For example, certain users at remote sites only require access to Asset data with low currency. It is determined that high-speed online access is not required; weekly transfer of Asset data to a local query system meets all requirements.

It is far easier to reconcile the needs of many user populations, together within the context of the organisation's efficiency, investment and information needs using a good Architecture than by allowing unplanned and uncoordinated Business Systems implementation.

Change management processes to deal with disaffected users also become less of an issue – a good Architecture encourages 'plug and play' within sensible boundaries. Obviously support and maintenance costs can heavily impact on this type of decision, but it needs to be considered, nonetheless.

# Business Requirements

Changes in Business process and goals, together with any resulting technology effects should be accommodated by your Architecture. A good Architecture will at least make the job of adjusting to change relatively straightforward. For example, if you choose ODBC as a standard, your flexibility and response to changing desktop application needs is enhanced.

Just as an Architecture can operate at a micro or macro-level, it will be influenced by macro-level Business requirements:

- The CIS must provide the customer with a single point of contact for the entire organisation.
- The business must provide customers with realtime fault repair information.
- A reduction in the cost of integration of (and later changes to) new Business Systems.

The latter point may be the result of experience. For example, in the case where the organisation is an amalgamation of other instrumentalities, the conglomeration of systems, data and processes was a burden whose impact was underestimated. The next time around, Executive Management wants an Architecture in place that enables Business System integration in a fraction of the time and at a fraction of the cost.

Equally, micro-level Business requirements will be very useful in defining Critical Success Factors for the Business and therefore, the Architecture:

- The efficient (Computer Aided Dispatch) of repair crews to the correct substation on receipt of Customer calls. The Customer ID, network connectivity model and tracing algorithms enabled the timely location of the fault.
- Using a common database, Customer mail-outs are prepared to notifying a power shutdown affecting both sides of the street to enable tree lopping to proceed. Tree lopping occurs for all affected customers within the stated timeframe, without inconvenient and expensive multiple power shutdowns otherwise occurring because the AM/FM/GIS customer connection data is not integrated with the CIS.

In other words, does the Architecture facilitate the easy solution of this type of micro-level requirement?

# System Integration and Distribution Requirements

System integration is in some senses an art rather than a science. As I suggested earlier, context is everything:

- Which Business Systems need to be integrated and why?
- How many users are View-Only, how many are Read-Write?
- Of the user populations, how many are 'Viewers', 'Power Users', 'Analysts', 'Data Maintainers' (and so on)?
- Do all user populations require up-to-date (current) data? For example, Marketing reports are run on aggregated Utility Network information once every 3 months. For the next Quarter the same aggregated data is used repeatedly for analysis and forecasting.
- How timely does the information have to be? For example, a user population requires accurate data within 1 minute rather than 1 hour.
- Do Power Users carry out some of the same tasks that the 'View Only' users do? Are they going to have more than one application (geospatial AM/FM/GIS and geospatial web)
- What *available* network bandwidth exists? Does anyone know?
- In the case of Satellite Offices, what are the data access requirements?
- How much data (volume) are involved? Where does it reside?
- How volatile are the data? Are there categories / classes of volatile (daily updates) or relatively non-volatile (yearly updates)?

There are many similar questions that can be asked. Prepare a matrix of user populations and various common requirements. Prepare a matrix of dataset characteristics (you can use this in your Corporate Information Model, which I will discuss later).

The task here is to have a sensible, objective and rational analysis of what actually *needs* to be done, rather than what someone thinks *should* be done

Thinking about the potential longevity of the Architecture is another important driver. For example, the risk of the RDBMS system becoming obsolete is low compared to a PC database reporting tool becoming obsolete (particularly if a major vendor decides that it wants the market share occupied by a smaller niche player).

Furthermore, it is possible to future-proof your system by avoiding the 'Gee Whiz' Factor, rather working with components that recognise industry trends and practices.

There is more to say about risk. I'll deal with some aspects of risk later in the paper.

# System Architecture with Tiers

I'll now spend a little time discussing System Architectures, because these are the means by which we see our Enterprise Architecture deployed to the end user.

Two tier architectures are pretty typical of conventional AM/FM/GIS systems. The application system accesses the data store – most query applications are intended for 2-tier environments (such as Seagate's Crystal Reports). Most applications can get by using a 2-tier architecture, the limitation being the number of concurrent users the architecture will support.

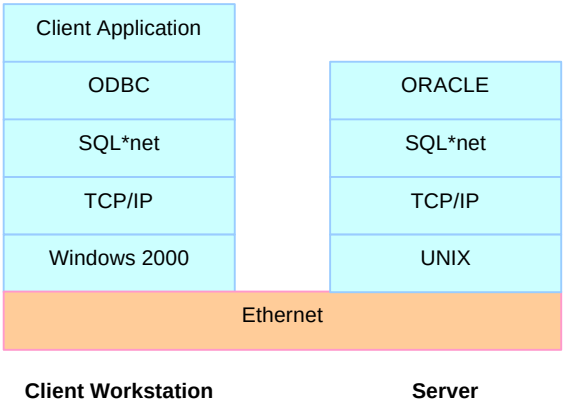


**Figure 2 – A Sample 2-tier Implementation**

A third tier adds the possibility of distributed processing, load balancing and database connection pooling (which reduces resource utilisation on the database server – hence concurrent database connections). The third tier can also provide the ability to distribute processing elsewhere besides at the client or the server.

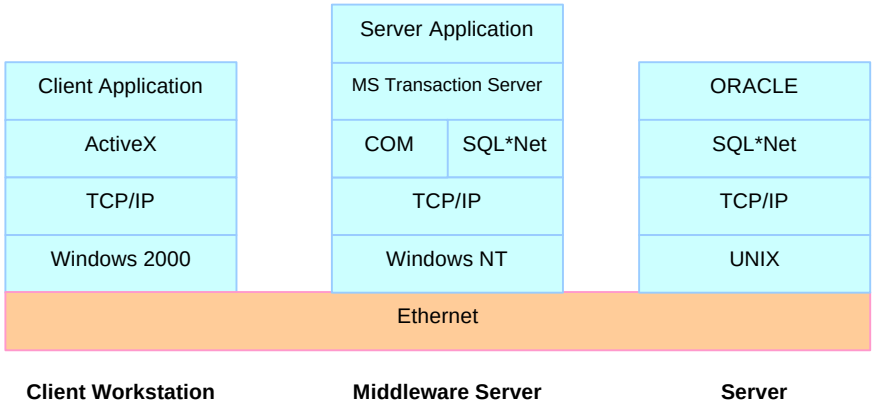


**Figure 3 – A Sample 3-tier Implementation**

Furthermore, if your AM/FM/GIS system is managing an intelligent, sophisticated relational database model, the third tier is an ideal place for the handling of tasks such as Long Transaction management, Business Rule processing and Data Transformation.

In AM/FM/GIS systems, the untapped potential of Web-based processing in one way leads us to 'n' tiers, where 2, 3, 4, 5 or 7 tiers are possible. This is not hard to do if you consider that Microsoft's DCOM offers a relatively homogeneous and coordinated approach to this type of complexity. First there is the ActiveX component that used to be a VC++ or VB component. This accesses an application server. The application server processes requests according to its business rules and requests combinations of geospatial and attribute information from the (remote) UNIX database server. The database object (table) may in fact refer to another database elsewhere on the network.

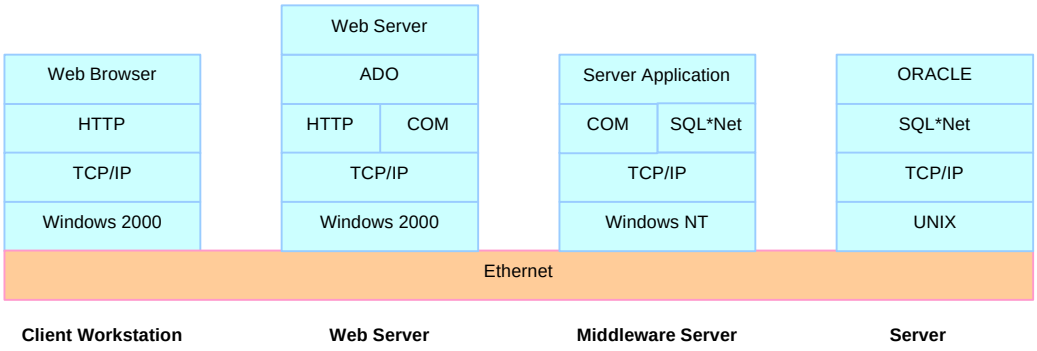


**Figure 4 – A Sample 'n'-tier Implementation**

All of these technology building blocks exist today. The bits in the middle are usually termed 'Middleware'. The issue these days is that the 'Middle' (as noted above) is becoming quite large compared to the other tiers. The 'old' 2 and 3-tier architectures can also be accommodated within this structure – the end-user should not notice any difference.

I believe that this type of Middleware environment will be common for many large enterprises within 3 to 5 years – there is no reason to expect that Utilities will be any different. The Web interface means that systems staff effort to install an application is minimal. You can even update the software 'automatically' from the Intranet or Internet should you so desire. Minimal end-user training is required, training costs are reduced. Minimal desktop hardware is required, so who cares if the PCs aren't 500Mhz Pentium IIIs with 128MB RAM? You have just made a saving on PC changeover that can support the cost of a new Middleware Server!

If you choose the Microsoft ActiveX / ODBC / OLE DB[2] route, you can provide sophisticated functionality through use of the same code base that supplies 'fat' client applications with their functionality. A user transparently executes the 'fat' code on a Middleware server from a thin client interface. DCOM provides satisfactory transaction throughput and latency through connection pooling and transaction management if you care to understand the technology and resource your Middleware Compute and Network layers.

You can choose to achieve similar results using Java based technologies such as Java, JDBC, Enterprise Java Beans (EJB), RMI and IIOP[3], to name a few (!).

Using this type of Architecture you can potentially provide your organisation with:

- A common infrastructure allowing many Business Systems to access geospatial data and intelligent model information in the case of Network Assets.
- Pooling of database connections – reducing concurrent RDBMS licensing costs.
- A superior ability to accommodate change – replace the Asset Number generation algorithm without impacting desktop applications.
- Better management of software licensing costs – this technology enables System Administrators to monitor *actual* concurrent user sessions
- Scalability – more Middleware processors can be added to the pool of compute resource.

[2] To start your research into Microsoft's OLE DB Technology, see http://www.oledb.com
[3] A good starting point for Java research is http://java.sun.com

This type of Architecture can reduce support costs for isolated application interfaces. The ability to release new applications in a timely fashion is improved because the effort required to dovetail new systems into the existing environment is less.

Now I have indicated elsewhere in this paper that a 'whole' to 'part' approach is a valid approach once you have defined the 'whole'. Armed with an Architectural Big Picture, that strategy is equally valid in the 'n' tier, Middleware world.

# Business Rules

As we move into the next millennium, an AM/FM/GIS system needs Business Rules in controlling the integrity of the network information model. I consider Business Rules as another type of "glue" because they enable the organisation's Architecture to retain integrity at the data level, and represent some encapsulation of corporate Business Processing.

Typically, Business Rules are enforced using Middleware or within the database. Conventional AM/FM/GIS systems enforce their own rules within their own environment. The rules are often not customisable in the same way (or to the same extent) that Business Systems require, so compromises or trade-offs are made to isolate the particular system from other Business Systems. File transfer, translation or transformation and aggregation are methods used to achieve integration with other systems and then only when really necessary.

In some cases, purpose-built software interfaces transfer specific subsets of data from the AM/FM/GIS system to other business systems. What is more likely is that these types of interfaces are used to import data into the AM/FM/GIS as the 'octopus' from which all geospatially-oriented decisions are made. This presupposes a GIS-centric view of the world, when what is needed these days is not geospatially-*replaced* but geospatially-*enabled* Business Systems.

In my experience, if Business Rules are not put in place there will be another, 'integration-negative' result: distrust of the source data, the corporate information model, and the data maintainers themselves. This is because the AM/FM/GIS oriented Business System runs the risk of capturing data in inconsistent ways, internally or externally, or with content that does not conform to the corporate information model. Distribution of data across the enterprise becomes problematic. Moreover, because of the 'bad press', Executive management will tend not to support the AM/FM/GIS system and its promoters, or form negative views of the valuable information that such a system embodies.

Of course, the use of Business Rules presupposes that you want an intelligent Network Asset Model maintained within the open corporate database for use by the rest of the organisation…

In the AM/FM/GIS context, Business Rules should apply to graphical editing and update as much as database relationships. For example, it might be desirable to prevent updates of attributes for which the Network Asset model maintenance team do not have authority to update.

# Corporate Information Model

Another type of glue that provides fundamental cohesion within an Architecture is a Corporate Information Model. This Model should be overseen by an appointed Corporate Data Architect.

The Corporate Data Architect provides the organisation with knowledge of *what* data it has (or wants), *where* the data comes from (its pedigree and quality), and *who* is responsible for the data. No matter where the information resides, if it is used by the Business its existence must be documented.

The benefit of documenting and maintaining such a 'data map' is that all users (technologists and users alike) better understand corporate data, which ensures that your company is more nimble in a Business sense and better able to exploit new Business Opportunities.

One obvious benefit is the increased sharing of data. The corporate intranet is an ideal way to advertise the existence of data within the Corporate Information Model. Of course, AM/FM/GIS data can be made available to other Business Systems.

Today, with the greater acceptance of georelational technology, geospatial data can be shared much more easily[4]. A Customer ID can be shared across Business Systems so that all users can know what kind of reporting will be possible. System integration also becomes easier – the integrators want to know where all these data elements physically reside, and in what form, so that they can realistically estimate how easy (or difficult) it will be to achieve integration.

I am not going to describe the formulation of a Corporate Information Model in detail. The topic fills textbooks but is quite a logical process (remember: *what* data? *where* does it come from? *who* is responsible for it?).

The Corporate Information Model for an organisation needs to be approached using the existing schemas from your Business Systems. Adding your geospatial data should be straightforward – this is not a physical implementation, you merely need to note what is where. It should be easy to identify what major entities each relevant Business System manages. Entities such as Customers or Assets are easily identified.

As I have suggested earlier in this paper, from a geospatial perspective you can merely identify the links to other systems, otherwise treating them as “black boxes”.

# Managing Data and Model Mismatches

When introducing integration between an AM/FM/GIS system and other Business Systems, you may have to consider rationalising both business processes. For example, one Business System may be charged with maintenance of certain asset attributes within the Asset System. The Network Management System may be charged with maintaining the model, geospatial accuracy and content for network engineering designs and as-built records.

Similarly, data rationalisation may be required – how does the unique geospatial ID match an Asset ID, for example – is a Power Transmission Pole identifiable in both systems. During this process you will discover mismatches – Ids that are VARCHAR2 (10) data types in one Business System may have to be mapped to CHAR (10) data types in another.

Further, is the granularity in both systems the same? What if the AM/FM/GIS system uses a single entity where the Asset system records more than one? For example, a network entity with sub-entities recorded as attributes. The sub-entities have Asset numbers. How do you maintain correspondence between the AM/FM/GIS and external Business Systems?

You then have to think about the values of common keys across systems. The unique ID of an asset recorded in the AM/FM/GIS system is rarely the same as that in the Asset system.

There is no way around this but dependent upon the capabilities of your AM/FM/GIS system, the georelational environment makes this task relatively easy, if tedious.

Obviously, the earlier this can be done, the better.

# Corporate Data Custodians

For an organisation to successfully achieve coordination between the schemas associated with Business Systems, Data Custodians are needed. The Data Custodian makes sure that the data is shared and integrated across Business Systems as appropriate. Ideally, there's one Data Custodian for each Business System, who ensures responsible use of that data across the enterprise. As a representative of a major corporate stakeholder (the respective Business System), the Data Custodian is also responsible for approving any major changes to the information model. The Data Custodians need to work very closely with the Corporate Data Architect.

[4] In the past, one unfortunate result of advertising the existence of geospatial information is that it actually took a lot of effort to translate and transfer data between proprietary formats, hence enthusiasm for data sharing was dampened. To some extent this problem still exists, in many cases because the effort remains or because of the cost of data to the purchaser.

# Transactions

The information technology world thinks of database updates in terms of transactions, rather than data fields. Another type of "glue" that I will briefly consider is the notion of Version Management. Using Version Management, an Architecture that includes the principles of historical as-built recording, in addition to future / proposed plans / designs (Job or Long Transaction Management), as well as point 'Records Correction' updates.

Many smaller AM/FM/GIS systems have been implemented as systems where data maintainers have direct edit access to the as-built database. For many AM/FM/GIS systems, a transaction model has not been considered at all. This makes it difficult to regard such systems as Business Systems, and this is another reason for the relegation of AM/FM/GIS systems to 'niche' status and a lack of true integration with other Business Systems, by and large.

If an AM/FM/GIS system has ability to manage plans / designs and updates, conventional systems often used a method that involved the use of Data Extract. This extract would be worked on for some period, than returned to the database (georelational or other) and 'burned into' the geospatial fabric of the database. The old data would be replaced by the new. Therefore, if a certain data element's value changed from 10C to 10B, then 10B to 11C, and finally 15C to 19D, only 19D would be sent to the database. The corporate database does not record the transition of the data element – it only sees the end-state. If the AM/FM/GIS system is using a georelational data store (as distinct from a georelational data model), Corporate Business Rules can be applied *only* to the end-state, if at all. Integrity is not ensured. The database must blindly accept any combination of valid and invalid state transitions passed by the application is a valid end-state. This situation may be acceptable for trivial attributes, but impacts on other Business Systems are likely, or integration with other systems is then made tedious. Since all information integrity is not managed by the database, there is the risk that internally consistent but globally inconsistent data is 'burned' into the database.

AM/FM/GIS systems that use a relational transaction model record all changes made to the entity – each has a Version. The application, the Middleware and the database are able to apply Business Rules to each state transition, as appropriate. Hence it is possible to:

- Roll back changes.
- Construct multiple 'what if?' designs.
- Track, monitor or audit the evolution of the network asset model.

Changes to the Network Asset Model that represent designs-in-progress can be accessed by authorised users only. Integration with Works Management Business Systems at a relational level is also possible, for example, as entities affected by the Long Transactions can be partially keyed by means of the Job ID from the Works Management system as a common, linking key.

Adopting an Architecture that includes a transaction model also means that you can avoid deleting the old data – old data merely changes states, and is not deleted. (It may be replicated to an historical database for future or offline reference.) This history is useful. For example, other Business systems can track the growth in value or extent of Network Assets based on this time-dependant (temporal) data. Management can track works in progress.

# Implementation Options

I've suggested that Architecture is an ongoing process of management and evolution. Some parameters will have greater longevity. Architectures will change. I will not tell you which Architecture is the best, or how to apply it. It depends on the context. I *am* going to try to predict the 'shape of things to come', present some scenarios and use some guidelines or commentary that will enable 'band-aids' to work without completely compromising a Big Picture Architecture.

## Short Term Solutions

Remember that a key to Business effectiveness may be the 'bandaid' for a short term. Consider the following true situation: An organisation needs quicker turnaround on 'approvals-to-proceed' with new work. At the moment, the task takes 3–5 days because of a combination of poor system integration and a requirement for a large proportion of inter-departmental dialogue, paper-work etc. The incumbent (GIS) vendor suggested a total system integration package amounting to hundreds of thousands of dollars and an implementation schedule of 12 months or more. The customer decided to spend less than $50,000 on

enough services and software (from another vendor) to provide an Intranet-based approvals system. The system was implemented within 3 months, integrated the mainframe and geospatial systems temporarily and apart from some minimal ODBC configuration, has longevity beyond that usually expected of a 'band-aid'. The Business Process now has a turnaround of 3 minutes or less.

The lesson here is that the customer was undaunted by the state of the internal systems. By using some preliminary, broad brush Big Picture thinking, along with immediate Business imperatives the solution was relatively easily accomplished – on time, under budget. The solution serves to engender Executive Support for further, planned and considered expansion of the embryonic Architecture.

# Longer Term Solutions

Utility organisations using technology will always have the need to introduce new innovations into their existing environment.

The Information Technology industry provides a wealth of options, and is providing innovation at a rate never seen before.

Outsourcing means that technologies underpinning Business Systems may not even be owned or controlled by the enterprise.

A further problem today is that the Internet has confronted us with potential that we cannot easily predict.

This means that formulating an Architecture must – more than ever – include models for future applications including the internet, extranet and intranet. During the TV documentary "Nerds" (ABC TV), Steve Jobs (CEO and Co-founder of Apple Computer) remarked that the Internet is a hotbed of innovation, precisely because Microsoft does not own it – on the Internet there is no one organisation controlling where you may want to go today.

As I have suggested, each organisation must analyse its requirements and use basic Architecture building blocks to develop and evolve an Architecture.

# Application Integration

AM/FM/GIS systems of the early to mid-90's attempted integration using customisations of the AM/FM/GIS GUI and API to simulate the user interface of a single application. This is not Enterprise Integration at all. It is a very large 'bandage' that merely expanded the role of the GIS user and ensured the dependence of the organisation on the GIS vendor, particularly where obscure 'niche' customisation languages were used. Whilst this sort of solution may work short-term, it will tend to obscure or postpone the need for true Enterprise (Business System) integration and the requisite thinking about Architecture.

These days it is equally common to use inter-application communications methods such as ActiveX to provide good native, even ad-hoc flexibility. The Architecture must therefore specify support for such methods, because integration through data exchange depends on each component's support for the chosen mechanism.

It is also possible for an AM/FM/GIS system to use the 'exposed' API libraries to achieve fairly tight levels of integration. But your AM/FM/GIS system must be compatible with the other application's language and architecture. As stated previously in this paper, interfaces built this way will create a hard-wired code between applications: there is a high probability that changes in one application affect the other. This reduces your applications' flexibility in the long term. I once worked on a system where the other application's API was changed within days of system delivery, rendering the interface useless much to the chagrin of both vendors and the annoyance of the customer. There was of course significant cost escalation to be borne by the vendors (in this case) and customer dissatisfaction.

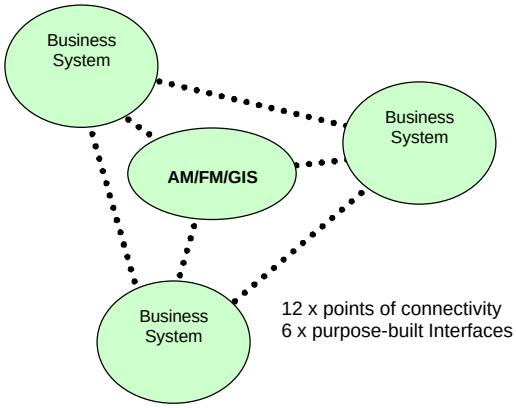


**Figure 5 – Interface Integration**

Web-based applications have made it possible to access data from diverse sources, because most vendors can at least agree that the Web is worthy of support! Because of a common tool base (Java or ActiveX plus supporting Middleware technologies) integration at the GUI level is a reality.

# Integration using Middleware

Using this level of integration, an AM/FM/GIS system may access corporate functionality available within the Middleware layer (that is, between the client and the server layers). For example, your network tracing algorithms might reside in common Middleware, rather than be defined in every geospatial application, using differing data across the organisation.

Middleware provides 'plug-and-play' at both the client and server side. This means that an organisation is not necessarily committed to one particular client application, or one particular database server. This of course scares the pants off both client and server vendors.

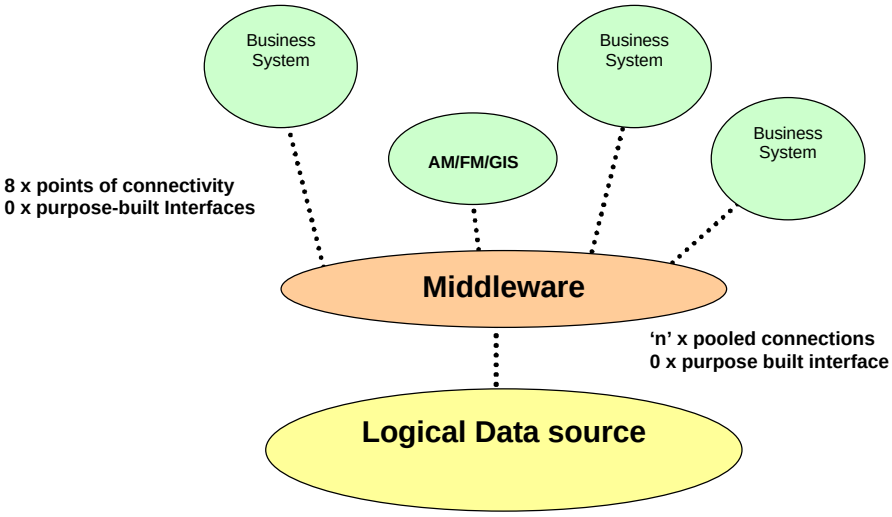


**Figure 6 – Middleware Integration**

The Architecture must define technology capable of providing this ability. This goes for the AM/FM/GIS system, in that it must support a connectivity method that is capable of accessing Middleware (such as ADO or ODBC). Both OMG's CORBA[5] and related Java Middleware and 'servlets' facilitate this type of implementation, as does the more homogeneous (but 'proprietary') DCOM[6] solution from Microsoft. Both of these technology sets provide good database integration and flexibility, and as noted earlier provide great distribution and integration options.

[5] OMG (Object Management Group) presides over the development of the CORBA (Common Object Request Broker Architecture), a vendor-neutral middleware architecture that establishes and manages client-server relationships between objects. See http://www.omg.org for more details.

6 More information regarding Microsoft's Distributed Component Object Model can be found at http://www.microsoft.com/data

Beware of database vendors that maintain that Middleware is somehow inefficient, and that the only way to integration nirvana is the creation of monolithic servers, where all functionality is embedded within the RDBMS kernel. I suggest that this 'marketecture' flies in the face of almost every piece of Distributed Systems and Component oriented thinking that has been articulated in the last 10 years.

Architecturally (objectively) speaking, distributed processing, data and Middleware *may* introduce a performance problem for short transactions. Alternatively, I have seen real-world, georelational applications run over a distributed system complete in a fraction of the time taken using a monolithic approach. The customer could not afford a bigger/better/faster monolithic database server. Distribution processing meant that any performance degradation was discounted by significant end-to-end execution improvements. As discussed previously in this paper, business context and individual needs are paramount to using technology. Middleware lets you scale the solution for the context.

The situation is not all roses. The Architecture needs to be cognisant of the fact that implementations of the same standards or Middleware from different vendors may not use the protocol in the same way (if you have ever tried to interface RS232C – 'standard' serial – hardware devices you will know what I mean). This has been a characteristic of CORBA implementations in the past, so that an Object Request Broker (ORB) from Vendor A can't easily talk to an ORB from Vendor B without major changes to the IDL (Interface Definition Language). Having said that, it is not too hard to find inconsistencies in the way that end-user applications implement the ODBC standard. And DCOM is arguably solely available on Microsoft's platforms.

Note that these Standards and Products very rarely support geospatial data types. For example, the ODBC standard does not include geospatial types or their spatial attributes. Vendors of geospatial systems that support ODBC Drivers have had to map geospatial types and attributes to conventional types and attributes for transfer using the ODBC Driver.

# Integration using Database Technology

At the back-end of an AM/FM/GIS system, there is a database. I am assuming that this is a georelational database – a relational database that includes some basic level of geospatial functionality or has been geospatially-enabled in some way through the use of intelligent Middleware. (I am not going to discuss databases made up of native – 'proprietary' – files in any great detail.)

I am also assuming an open, transparent physical schema for georelational data, modelling entities and relationships – in a way that closely resembles the way that relational technologies have enabled businesses for the last decade or more. Here, geospatial entity versions are managed as rows ('tuples') within a relational table. Spatial location is merely another entity attribute stored within the table (or perhaps in another table, due to site, model or performance specific reasons).

As noted earlier, it is possible to provide a 'band-aid' solution to meet short term Business imperatives. One method of dealing with an island of AM/FM/GIS data is to replicate a snapshot from the AM/FM/GIS system to the Corporate RDBMS for use by other Business Systems using interfaces such as reporting tools and Web viewers.

The move to a georelational world provides huge Architectural benefits. If an AM/FM/GIS data is georelational, transactions to another target database can be triggered based on changes to the 'as-built' model. For example, all superseded entities may be moved to a temporal database. This transfer might include some form of in-process aggregation, summarisation or generalisation.

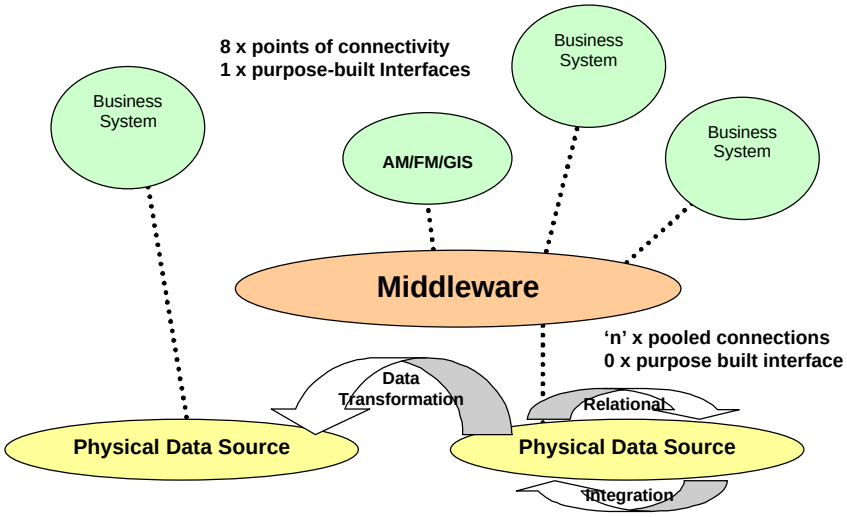


**Figure 7 – Database Integration**

Alternatively, an AM/FM/GIS database can be partitioned into Volatile and Non-volatile data. It may be that the Network Asset Model can be managed using a database server and Middleware distributed at Departmental level. Model maintainers operate using a volatile environment based on the Departmental Server. At the completion of a long transaction, updated 'as-built' records are replicated in-process to a relatively non-volatile Enterprise Server for use by other Business Systems.

So what if your AM/FM/GIS system uses a different RDBMS? Every major vendor supports Gateways to ensure this sort of process is possible. For some it is as easy as configuring ODBC Drivers. For others the inclusion of what are effectively in-process stored procedures takes care of data type mismatches.

In summary then, you should get your AM/FM/GIS data accessible by Middleware and Web clients as soon as possible. So many options are possible when the data is managed within the georelational paradigm.

Finally, there is a Holy Grail to be dealt with here: the concept of all data stored in one database conforming to one model. I hope that I have demonstrated that this may not even be useful, or if so could be a waste of time and money because of other Database, Database Connectivity and Middleware options. The trade off is about time, technology, effort and real Business Needs. I suggest that it is only with the benefit of a Big Picture – an Architecture – that piecemeal tradeoffs are low risk.

# Skills

Skills are a type of "glue" that provides cohesion to the overall Architecture. Organisations need to attract and keep staff having the right skills. Technology evolution over time means that the possible content of Architectures has changed over time.

These days, various core technologies from a few large vendors are often implemented. Skills can then become commodities – except for the fact that quite often the system integration in *your* context requires your staff to be cognisant of the business (your mandate). It will not be possible to hire contract staff for all Business System integration tasks.

In some cases Business Systems are outsourced together with the skilled staff. Again, *your* staff in *your* context requires cognisance of the business (your mandate). You can't outsource core business knowledge.

I have seen a number of organisations whose AM/FM/GIS systems are totally dependent on a few individuals who actually know how any Business System Integration works. Objectively speaking, this is not a good position for an organisation to be in.

The rate of technology change also means that the technical skills required to design and administer this environment are different. The web-based technology base alone provides a whole new sphere of skills that can't be mastered quickly, ranging from Java to VPNs (Virtual Private Networks).

Developing applications is now easier, but potentially more difficult to manage and troubleshoot within a distributed environment.

Moreover, Time-to-Market pressures on the software vendors means that Customers have been provided with products that are effectively in Beta test mode (such as certain unreliable versions of Browsers from both Microsoft and Netscape within the last 3 years) resulting in unexpected issues for an organisation.

Skills are critical, and organisations need to foster a different, component/object oriented mindset in staff to cope with the Architectures of today.

# Risks

Sensible and objective risk analysis means that an Architecture can include innovation, because the guiding parameters control the introduction of innovation. An Architecture need not be hamstrung by a 'buy what everyone else has' mentality because the Architecture uses Big Picture principles. The Architecture should include the parameters within which technology can be prototyped, perhaps in response to a need for an urgent 'band-aid' solution. Proven performance in this environment provides comfort:

- Does the application work as advertised?
- Is the performance acceptable and does the application scale?
- Is integration with other Business Systems easy or hard?
- Are the required maintenance skills and effort prohibitive?

Correct expectations, specifications agreed by all stakeholders, formal management reviews and reports, bound together with rigorous and objective project management will help the organisation to integrate new technology into its Business Systems portfolio. The same applies to the upgrade of existing systems.

Otherwise an Architecture can only ever use technology that is dispensed by the ever-decreasing set of large vendors, which negates the nature of technology and its innovation. If you attend product unveilings by the large IT players with any frequency, read between the lines: "you can have any innovation that you want, just so long as it is our innovation".

As stated earlier in this paper, one large risk is the unknown, particularly based on the burgeoning world of Internet technologies. The problem is related to "not knowing what you don't know" or being 'blindsided' by some unforseen impact. For example, I recently heard an IT Manager indicate that his (understandable) concern was for the vast majority of his users that had no geospatial systems access. Internet technology is a great way of enabling these people's workplace.

However, what is the systems impact of an exponential change in usage patterns and a groundswell of increasing user population driven by the corporate Intranet? There is a 'feeling' that it will be considerable. There is also a risk that the Business will expect more of the geospatially-enabled systems than is initially envisaged. Sensible connectivity, data, database and Middleware standards within the Architecture will help 'future-proof' the organisation against the unknown profile of this expected growth.

# Executive Sponsorship

Executive-level support for a corporate Technology Architecture is mandatory, and another form of "glue". Assuming that the need is understood, the Corporate Executive needs to appoint an Executive Sponsor for the Technology Architecture to ensure that it has a champion.

In this way the implementation of a 'Big Picture' Architecture will be provided with a top-down interest in matters of budget and long term focus (together with discretionary power to remove road-blocks), and that is all a contribution to the Business. Results get support. But you can't get results without support. Both top-down support and bottom-up reporting are required. It's a relationship where there is tangible commitment to business success from both roles within the organisation.

Executive Management may not understand the need for a formal Architecture. In a competitive environment, ask yourself which organisations will have the advantage. Assuming a level playing field, it will be the organisation that makes best use of its information. And geospatial information may provide an edge. The task is to realise that advantage:

- Demonstrate your AM/FM/GIS data using new IT initiatives – web interfaces are ideal.
- Start with a simple list of who owns what data, where it is and what it represents. Use the corporate Intranet to publish this information.
- Lobby for a full-time Corporate Data Architect. This person will begin to build and maintain the Corporate Data Architecture.
- Appoint Data Custodians for each Business System, who work with the Corporate Data Architect in a Team to keep all Business Systems working cooperatively.

# Conclusion

Creating your organisation's Technology Architecture requires the commitment of the organisation and the individual. It is a Team Effort. Executive Sponsorship is essential. Good, clear thinking is required. The Business imperatives must drive the basic presuppositions.

A Big Picture is all-important. Develop it first, add necessary detail at need. If your organisation is to acquire / upgrade geospatial technology, spend time to understand your Architectural requirements.

You may need to employ someone to 'mentor' your formulation of the Architecture, or work closely under your supervision to achieve a result.

The end result will be the Big Picture, the 'whole' with which technology and integration 'parts' can be chosen, coordinated and implemented over time, with reduced risk, reduced cost and increased flexibility and innovation. Most of all, you'll likely acquire technology that is aligned with business needs, and can move with the changes in those needs.