

The Inspired Enterprise Architecture Frameworks©

An *Inspired* white paper by Graham McLeod, Managing Partner, 2002

① www.inspired.org ✉ mcLeod@iafrica.com

Introduction

Enterprise Architecture embraces the disciplines of assessment, visioning, design, controlled evolution and improvement with respect to business, applications, information, technology infrastructure and methods and practices. John Zachman proposed a framework for the field many years ago. He suggested the following dimensions (columns) and levels of abstraction/detail (rows):

	DATA	FUNCTION	NETWORK	PEOPLE	TIME	MOTIVATION
SCOPE	Identify Entities	Identify Business Processes	Map of Business Locations	Identify External & Internal Agents	List Significant Events	Business Goals and Strategy
ENTERPRISE MODEL	Entity Relationship Model	High Level Process Flow Diagram	Logistics Network	Organization Chart	Master Schedule	Business Plan
SYSTEM MODEL	Attributed Data Model	Data Flow Diagram	Distributed System Architecture	Human Interface Architecture Function<=> Person Role	Processing Structure	Knowledge Architecture
TECHNOLOGY MODEL	Relational Model	Module Structure Chart	System Architecture	Human Technology Interface	Control Structure	Knowledge Design
COMPONENTS	Database Schema	Program Source	Network Architecture	Security Architecture	Timing Definition	Knowledge Definition
FUNCTIONING SYSTEM	Database	Program Object	Network	Organization	Schedule	Strategy

Figure 1 Coverage of Enterprise Architecture and Techniques for Analysis - John Zachman

Over several years, the framework was filled in by suggesting techniques and associated deliverables for the cells in the matrix. These are illustrated in the accompanying figure. The suggested techniques were adopted from popular approaches to specific areas prevalent at the time. While intuitively appealing, this approach was not necessarily practical or easily achieved.

Difficulties we observed in organizations attempting to get to grips with EA include:

- The large variety of techniques required a large set of skills to apply. These skills were seldom all available, and very seldom in single individuals or small teams. Thus, architecture work required the assembly and coordination of large numbers of people
- Integration of results delivered in the various techniques was non-trivial. Many of the models interpret similar concepts differently, and provide overlapping perspectives, but not necessarily at the same level of detail/abstraction
- Management of the information gathered and the variety of models, cross references and associated documents could become an administrative nightmare

In doing strategic architecture work with our client base over several years, we evolved a different but complementary approach. We believe that the Zachman framework provides a

useful agenda - i.e. A focus for what we should be thinking about, but we felt the modeling and integration could be vastly improved. Being from a systems background, we analysed the meaning and usefulness of a framework, synthesised meta models to represent and manage this, and then proceeded to populate the framework with suitable content. In parallel, we evolved workable processes to follow in practically implementing an architecture culture and way of working in an organization. In summary, our approach provides:

- A **framework**, which identifies the kinds of things that should form part of the architecture and their relationships, interfaces and standards. We have specified these for business architectures, application architectures, information architectures, technical architectures and methods architecture. There are links to strategy and project and program management, as well as to software architecture and system delivery methods.
- **Criteria** for elements in the framework which, together with requirements, allow appropriate choices for each element
- “**Short lists**” of good options which can provide interim guidance until full architecture analysis has been achieved
- Architecture implementation **processes** to assist in adoption of the approach
- The Archi **knowledge management tool**, which has been populated with the frameworks and a meta model representing all the important architecture elements and relationships. This provides a repository and web based access to support a community of architects and other interested parties

The balance of this paper is mainly concerned with the Inspired Architecture Frameworks.

Key Concepts

Framework

We consider a framework as defining the types of **parts** and their **relationships**. For a building, the types of parts you would commonly expect would include: roof, windows, door, walls, floor, foundation etc. These have specific relationships and interdependencies: e.g. The roof is supported by the walls. The door passes through the door, etc. We talk about these parts as architecture **elements**. Elements in an enterprise architecture framework would include: business units, application systems, products, technical infrastructure components (such as platforms) and more.

For each element in the framework, there are various available **options**. For the building, we could choose a floor of wood, concrete, tile etc. Which is the best option to choose will depend upon our goals and **requirements**. If we are building a factory, for example, our major requirements might be durability and cost. If we want a luxury house, maybe aesthetics and comfort are more important. If we want to design a hospital, safety and ease of cleaning will be very important. We can also devise a set of **criteria** relevant to each element. For a floor, these might include: Load bearing capacity, Aesthetics, Ease of cleaning, Cost, Durability, Safety. We can consider each of these characteristics for available options in the light of our requirements. You could say the requirements provide relative weightings for each criterion. These concepts help us to choose the best option from those available for each element. When we have chosen an option for each element and populated the framework with these choices, we refer to the result as an **architecture**.

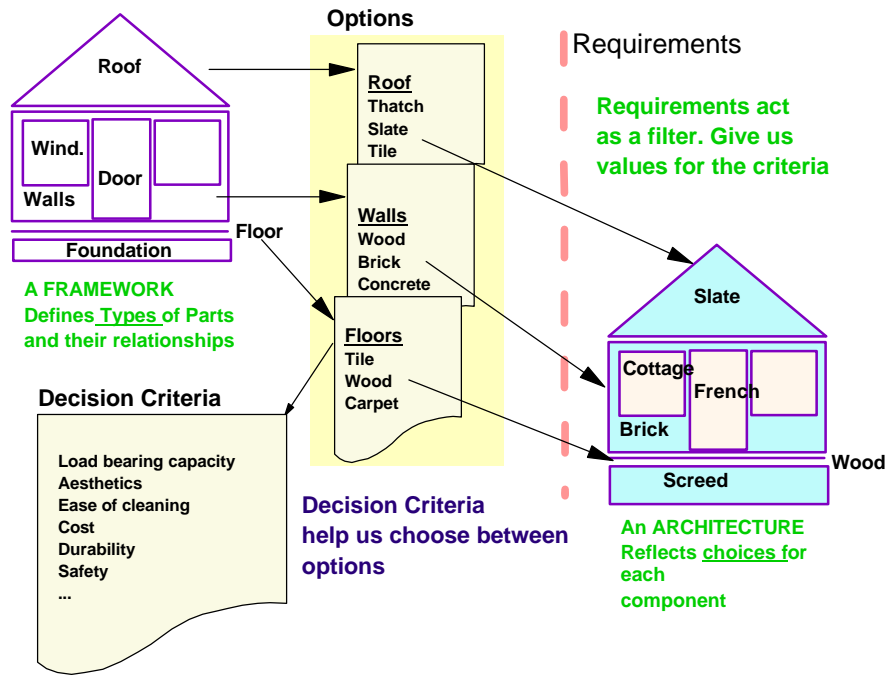


Figure 2 - Framework and Architecture Concepts

Interfaces

A further consideration when choosing elements is their compatibility with each other. For example, we do not want to choose a heavy slate roof if the walls cannot support it. In an IT architecture, similar considerations exist between applications and platforms, communications infrastructure and database technology. We will thus be careful to consider which elements interface or interact with which others and how. We need to consider relevant standards, which can ease the integration of the various elements.

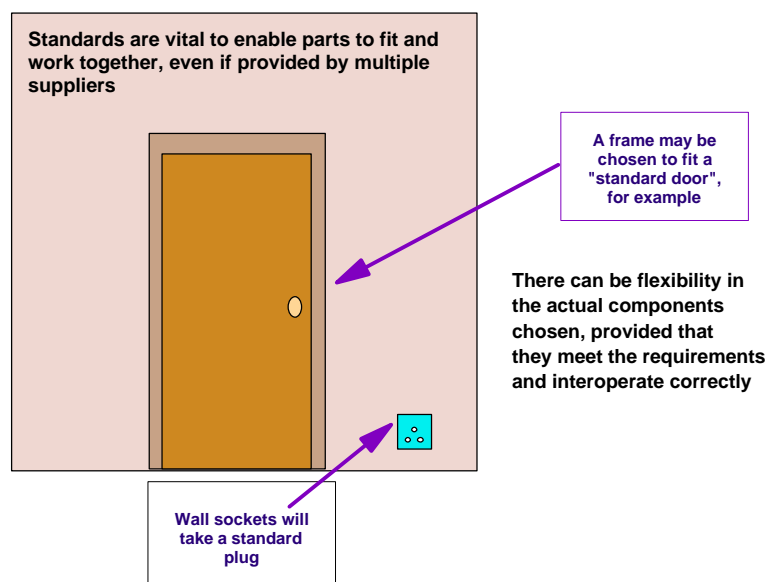


Figure 3 - Components, Interfaces, Standards

Present and Future Architectures

Some may doubt that they have a current architecture since they feel it just “happened” as opposed to being designed. There is always a *de facto* current architecture, even though it may not be pretty. We can at least document what it looks like and analyse its effectiveness - kind of like drawing up a building plan after a house has been built. You will still find this very useful if you are planning any major changes.

The future architecture is a goal to aim for and may never be achieved in full or as currently conceived. What it does do is allow us to move in the same direction with a shared vision and to debate the suitability of that goal and design. The desired future will, of necessity, also evolve as we discover new issues, opportunities, technologies and business models. We may develop several future scenarios to evaluate their respective merits.

We see strategy and the architecture management process as moving from where we are now to where we want to be with a minimum of risk, effort, cost and disruption.

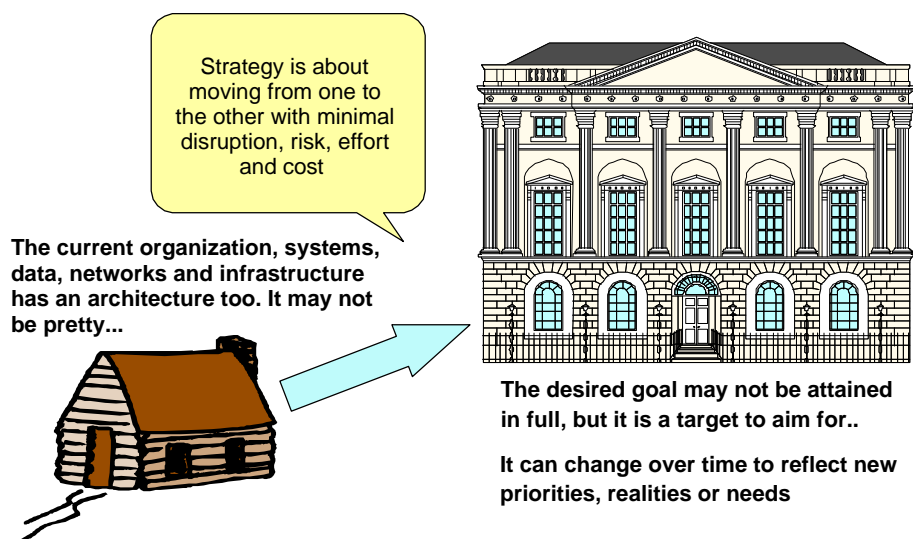


Figure 4 - Current and Future Architectures

The Inspired Enterprise Architecture Frameworks©

The Inspired Architecture Frameworks provide a model for the capture, management and evolution of knowledge related to an Enterprise Architecture. They allow the capture and management of information related to:

- Business Architecture
- Application Architecture
- Information Architecture
- Technical Architecture
- Methods Architecture

In addition, they provide several supporting types for things like architecture requirements, business drivers and choice criteria.

The frameworks are provided at two levels:

- Hyperlinked visual models suitable for presentations, training and navigation aids (some of these models will be reproduced in this paper). These primarily provide a conceptual and logical view of the frameworks.

- A formal meta model which details all the architecture elements, their relationships and attributes as well as linked knowledge assets, such as documents. These are suitable for the building of repositories, models and formal management of architecture elements. They are available preloaded in the Archi Knowledge Management tool. More on this later. The formal model comprises some 80 plus types and over 300 relationships.

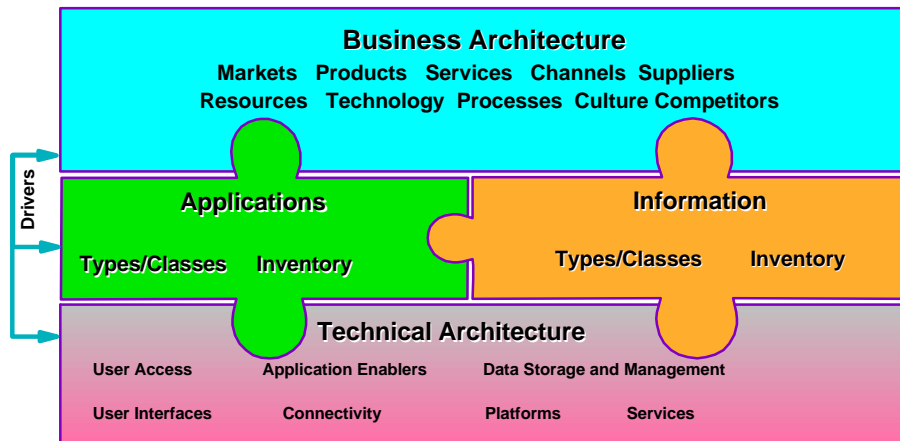


Figure 5 - Inspired Enterprise Architecture Frameworks - Top Level

The above picture represents the four major dimensions of the architecture.

- The **Business** component deals with the enterprise and its environment, as well as the architectural elements within it.
- The **Applications** component is concerned with the (computerised) systems that support the enterprise. These make use of information and run (execute) on a variety of technical platforms.
- The **Information** component provides support for business decision making as well as to the applications. At the logical level it represents logical subject areas of related information, e.g. customer information, product information, services information. At the physical level, it includes various databases which reside upon platforms in the technical infrastructure.
- The **Technical** component consists of the hardware (computers, storage devices, input and output devices), networks and operating software that support the storage of information, processing and communication of information throughout the enterprise, as well as into and out of the enterprise.
- We can also visualize a **Methods** component, which details the way the organization goes about creating strategy, managing architectures, delivering systems, managing projects and the operational environment.

A further consideration in building a future-aware architecture, is the general future environment in which the enterprise is likely to operate and the impact of this on the shape and requirements of the future enterprise. This is detailed in the **future world scenarios**. Here you will find information related to issues such as:

- Economic environment
- Legal and regulatory environment
- Social environment
- Technology landscape

These are discussed with respect to how they may influence the customers, products, services and organization of the enterprise and what architecture implications there might be.

Business Influences Architecture

Architectures must respect business goals and values. These goals will shape the way in which we will make decisions when selecting architecture options and components. Our choices may be very different depending upon whether we want to optimise cost or time to market, for example. The matrix shown is a fragment from this type of analysis for a telecommunications provider. Through understanding the business drivers, we can put in place architectural guidelines which will shape decision making.

Business Driver	Cut Costs	Efficient One Stop Service	Shorten Product Intro Time	Grow Market Share	Good Corp. Citizenship	Focus on Core Competencies	World Class Service
Architecture Req.							
Lower TCO thru Standard Config's	✓		✓			✓	
Reduce # of technologies per arch. component	✓	✓	✓			✓	✓
Data and network integration		✓	✓	✓		✓	✓
Follow internat'l standards	✓	✓	✓	✓	✓	✓	✓

Figure 6 - Business Drivers for Architecture Requirements

Technical Capabilities Influence Business

The reverse side of the coin is that emerging technologies can create business opportunities. By considering innovations relevant to our industry, we can identify opportunities for business innovation that can be supported by the technology. Sometimes the new approaches can offer whole new business models.

Technology Driver	Massive Cheap Bandwidth	Powerful Portable Devices	Affordable Digital Signal Processing	Cheap Reliable Digital Storage
Business Innovation				
Free Local Access	✓			
Video Conferencing Service to Homes	✓		✓	✓
Voice Commands for Svc Transactions		✓	✓	
Keep all documents electronically	✓	✓	✓	✓

Figure 7 - Technology Drivers for Business Innovation

Business Architecture

The types in this section are designed to support the modeling of an enterprise, the context within which it operates, and its interactions with parties in this environment, including customers, partners, shareholders etc.

The types cover issues such as: Customers and Markets, Products and Services, Channels, Organizational Units, Deals (contractual or financial arrangements between parties), Business Processes, Culture, Competitors, Business Goals and various other aspects.

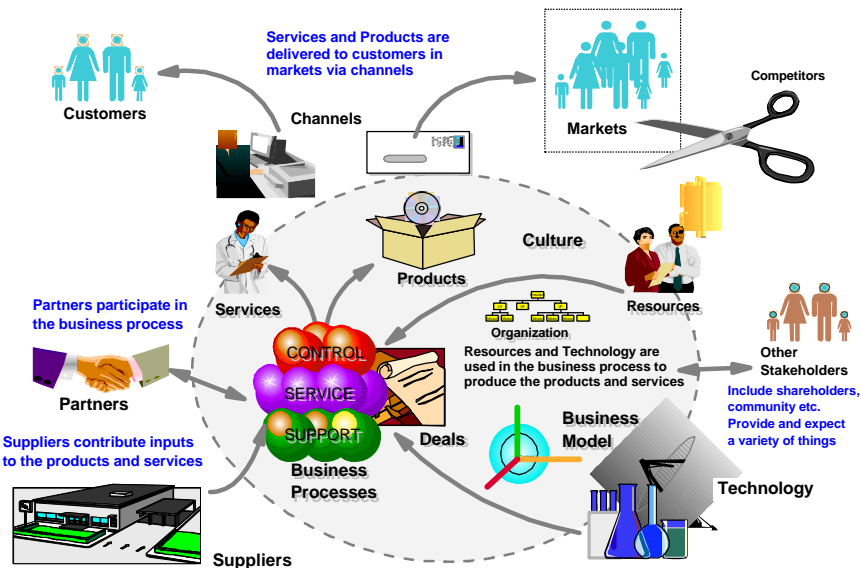


Figure 8 - Business Architecture

Customers provide revenue and ensure the future of our business. They should be the focus of every change. Satisfying and delighting them should be the focus of each business process.

Markets are a way of grouping customers with similar needs or profiles. They allow us to assess the relative size and importance of various groupings and categories. They also allow us to devise and manage products and services to satisfy various types of customers. We can see which groups are growing or declining relative to others, thus preparing us to address future needs.

Competitors are always there and ready to take over markets that we ignore or where we fail to offer an attractive product or service.

Services and Products are delivered to **customers** in **markets** via **channels**.

Channels are ways to reach customers. They may include, for example, retail stores, partners, mail order and the Internet.

Suppliers contribute inputs to the products and services. Business processes transform these inputs to the products and services which we supply to customers.

Deals are the contractual or legal arrangements that we hold with other parties. They detail who contributes in what way, who benefits in which way and the respective responsibilities and roles of the participants.

Business Models show how we interact with various parties, how we profit. They may extrapolate over several years based upon various scenarios of likely futures.

Resources and **Technology** are used in the business process to produce the products and services.

The organization has a **culture** which shapes its vision, structure and the way it deals with people. Culture is about the enduring values and principles which the organization uses as a background to decision making and behaviour

Partners participate in the business process. They may perform parts of it in a wholly independent way. We may form part of the business process of other partners.

Not reflected in the above picture, but part of the business architecture, are the following items:

- Business Objective
- Business Object (which links to the Information Architecture)
- Business Innovation
- Business Goal
- Business Function
- Business Event
- Business Communication
- Brand
- Business Rule
- Business Unit
- Budget

There are several lower level architectures to which we can “drill down” from the business architecture, including ones for business processes, products and deals.

Application Architecture

Applications are computer systems applied to the solution of business problems or realisation of organizational or personal goals. They often directly interact with and support the work of end users. Typical examples would include a finance system, debtors system, order processing, bank account management, personnel etc.

Application Types are likely to change over time and the mix can dramatically change the required infrastructure. Our Architecture should reflect these changing needs and opportunities. Some trends include:

- Increasing use of Multimedia and rich data types (audio, video, graphics, animation)
- Decreasing use of batch sequential processing, increasing online and realtime processing

- More analytical processing (Decision Support, Data Warehousing/Mining, Parallel Processing, Multidimensional matrices, etc.)
- More document handling (text, rich text, scanned documents, OCR)
- More automation of business processes, workflow

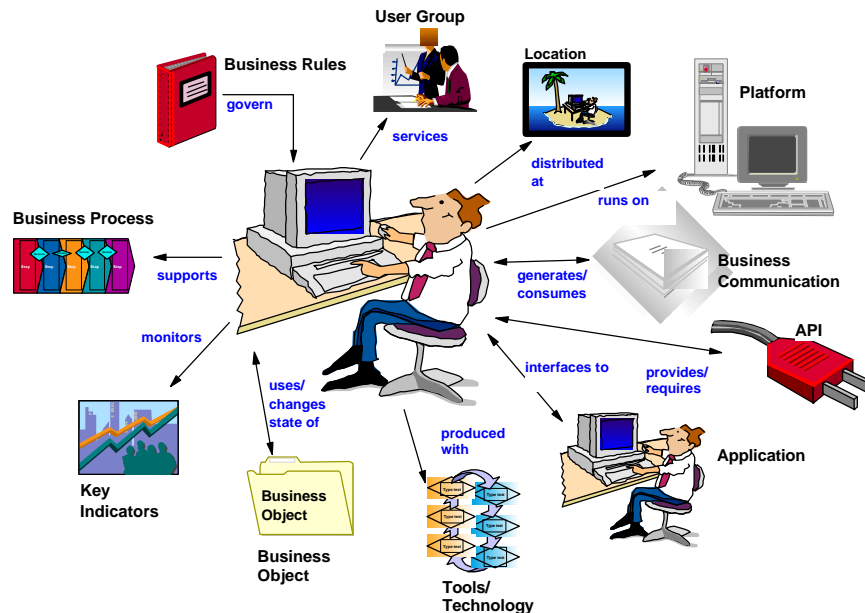


Figure 9 - Application Architecture

Classes of Applications address different business needs. We can determine what the business has in various functional areas, thereby highlighting deficiencies which are likely to result in future projects. In other cases, we may find that different business units have conflicting solutions for the same need. This can represent wasted effort or resources and may complicate cooperation and integration. Different classes of application can be examined for their respective contribution to the particular type of business and this knowledge used in prioritising projects.

We typically build an **inventory** of applications, detailing their characteristics and relationship to the rest of the architecture. Alongside the current choices, we can express recommendations for the medium and longer term. This gives a sense of what will be retained, evaluated, implemented or phased out.

We normally build an **Application Inventory** which identifies what applications we have as well as those that we are putting in place via projects. The inventory allows us to see where there are gaps in our portfolio as well as allowing us to identify opportunities for better utilization or rationalization across business units. It also allows us to gauge whether our support infrastructure is adequate and to see how our applications map onto the technical environment and relate to the databases in place.

Selection Criteria help us to evaluate a potential package, vendor proposal or inhouse development proposal. These can assist us in making good choices which fit with the rest of the architecture and are strategically sound.

Once new applications are added, selected or developed and are to enter production, we will want to add them to the inventory. There are templates provided for this purpose.

Information Architecture

Information is fundamental to the successful operation of a business.

We need accurate and timely information to do the following:

- Determine profitability and track progress towards objectives
- To monitor quality and service delivery
- To support decision making at every level: operational, tactical, strategic
- To prevent fraud and abuse
- To optimise use of scarce resources
- To make most efficient use of assets and resources
- To create and deliver sophisticated products and services
- To manage people and groups effectively and grow their capabilities and prospects
- To interact successfully with our suppliers, customers and business partners

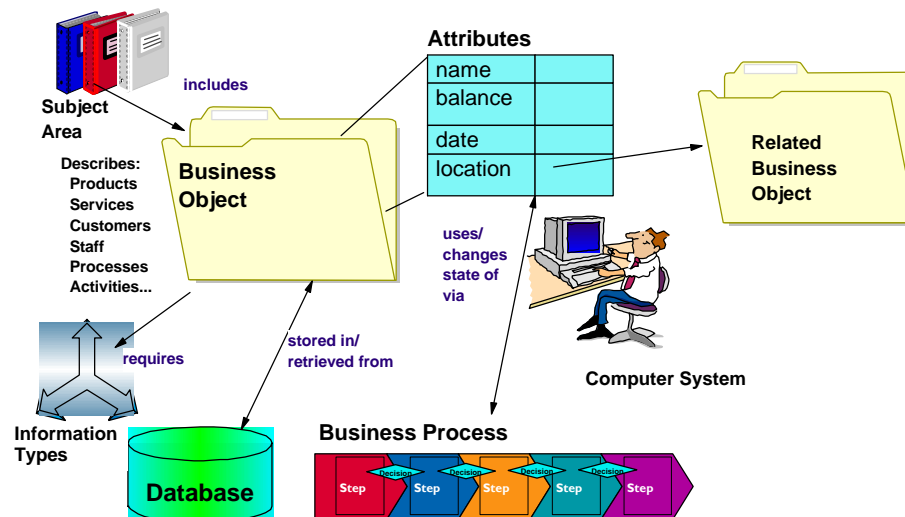


Figure 10 - Information Architecture

In building architectures, we are interested in information in several ways:

- **Types** of information are important as they have significantly different capture, storage, processing and transmission requirements
- **Subject areas** provide a high level view of related sets of information (e.g. Customer, Product, Organizational Unit, Supplier, Order, Human Resources, etc.)
- An **inventory** allows us to represent what information we currently have and how and where it is stored, as well as how accurate and timely it is
- **Business Objects** and their relationships provide a business domain (logical/conceptual) view of the information
- **Databases and data collections** provide a more technical/implementation (physical) view of the current information assets
- We also need **criteria** by which to judge whether a particular information collection has been well designed and implemented and could be usefully incorporated into our overall information base

- **Relationships** to application systems and business processes link information to other facets of the architecture

Technical Architecture

The technical architecture provides a framework for specifying the technology elements of the organization's infrastructure. It includes: Platforms (hardware and software combinations supporting execution of applications), Networks, Data Storage and Management, Security, Internal Software Architecture of Applications, Middleware, User Interfaces, User/Function Interaction Models and Development Tools and Environments.

The technical architecture is normally described at the level of *types* of things, rather than instances. This is to avoid the problem of excessive detail bogging down the architecture process and strategy. It is possible to track the technical elements at an instance level if desired, but this requires considerable commitment, effort and dedicated resources.

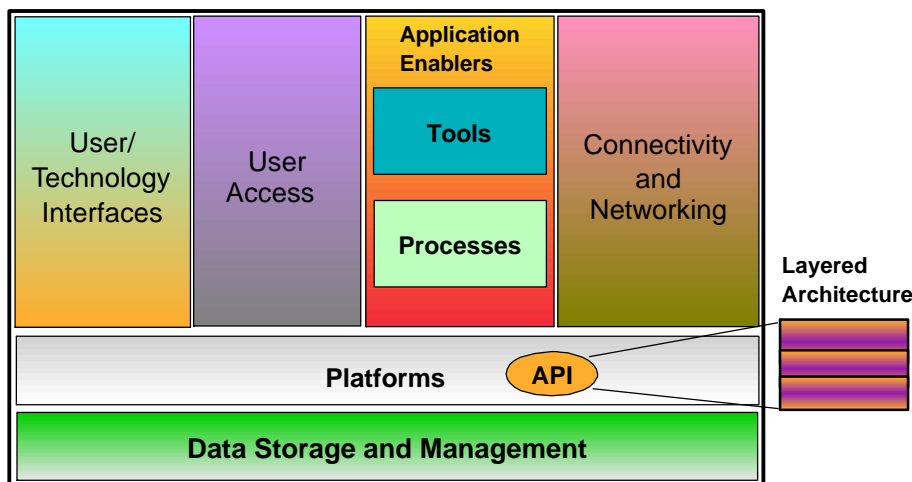


Figure 11 - Technical Architecture Overview

The above diagram summarises the technical architecture elements. They include:

- **User/Technology Interfaces** which includes the devices which link user to application systems and other technology elements, as well as standards for representation of data on these devices.
- **User Access** refers to the relationships between types of users and functions provided by applications, geographic distribution of facilities, physical access, adaptations for special needs and mode of interaction. The latter identifies what means users will use to access the various functions: e.g. text command, text menu, graphical user interface (GUI), speech, etc.
- **Application Enablers** deals with the Processes, Techniques and Methods used to create applications and the Tools supporting application delivery (e.g. languages, compilers, CASE tools etc.)
- **Connectivity and Networking** refers to all elements which connect hardware and software elements. It includes network links, switches, network protocols, network management software and the like
- **Platforms** refer to the unique types of combinations of hardware and system software upon which application systems run. An example would be an Intel processor with the Unix operating system

- **Data Storage and Management** relates to all elements necessary to store data persistently and to retrieve it rapidly and reliably. It includes hardware devices such as disks, tapes, optical devices; as well as software elements such as database management systems (DBMS), dictionaries and the like
- **The Application Programming Interfaces (APIs)** refer to the interfaces by which applications make use of services on their execution platforms
- The **Layered Application model** refers to how an application should be partitioned to allow optimal distribution across a heterogeneous and multi-tier deployment environment. This may be self contained in one location or distributed across a local or wide area network

All of the above elements have further detailed architectures in the full frameworks.

Methods Architecture

The methods architecture details the processes, techniques, deliverables and resources used by the IT organization itself in meeting the needs of the business. It will typically include project management, system delivery, change management, architecture management and capacity planning in its scope.

The types provided support a generic method modeling capability suitable for the representation, evaluation, evolution, integration and management of methods, compliant with the McLeod Method Model. Archi has special support for this model and can serve as a method engineering platform.

	Strategy	Analysis	Design	Construct	Deploy/ Support
Process Improvement	Capability Maturity Model (CMM), SPICE				
	Architecture Management				
Project Management	MITP				
				Change Mgmt	
Development Methodology	Stakeholders, Value Chains, CSF's	Object Analysis, Package Selection	Design, Component Assembly	Test	Customer Support
Techniques	ROI, Comp. Adv., BPR	Object Modeling, Bus. Proc. Modeling, Event Modeling	Prototyping, Perform.	Model, View, Controller	Download, Change Control
Tools	Strategy WB	CASE, Libraries	Visual Construction, Components	Tools	Deploymt Tools

Figure 12 - Methods Architecture Overview

Links to Project and Program Management

Architectures remain theoretical if they are not a vehicle for realising business goals. They also need to play out in the practical decisions made in technology and application acquisition, as well as projects and programs. In the Inspired approach, we perceive the link between strategy, architecture, project and program management to be vital. This is illustrated in the accompanying diagram.

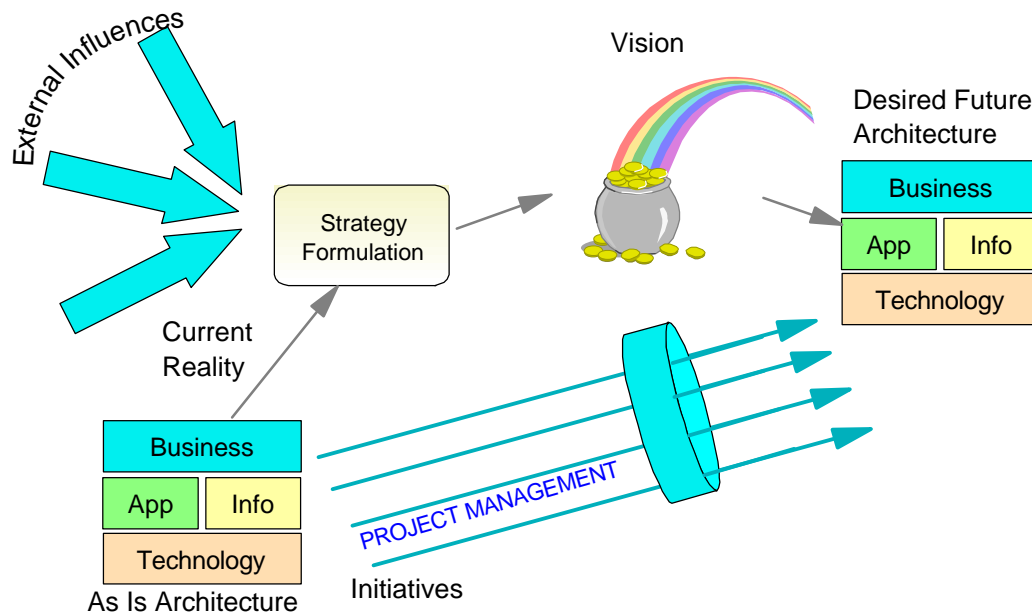


Figure 13 - Relationship between Strategy, Architecture, Program and Project Management

Strategy is shaped by external influences and current reality. Current reality is largely represented by “as is” architectures. First a vision is formed, then this is expressed in more detail as a desired future architecture. Projects and initiatives are mounted to move from the current picture to the desired future. Program management is represented by the collar around the projects - ensuring alignment and congruence, as well as allowing sharing of resources and sensible prioritisation.

Links to System Delivery

It is also vital that system delivery projects take cognisance of the architecture. We encourage this by using “delta” models at project definition and design stages. These involve extracting relevant details from the current architectures and considering the net change (delta) required for the new initiative. In the process we highlight reuse of existing elements, necessary adaptation of existing elements and new elements that will be required. The approach emphasises reuse of existing assets, coordination between developers and architects and identification of common requirements and solutions.

Example of Selection Criteria

Earlier, we mentioned the need for criteria to help us choose good options for architecture elements. The frameworks provide guidance in this respect in three major categories:

- Choice of applications
- Choice of Information resources
- Choice of Technical components in the infrastructure

An extract from the selection criteria for an application is shown as an example.

Application (1 of 4)			
	CRITERION	COMMENTS	REFERENCE
1	Consistent with Business Objectives		Business Architecture
2	Data Resides in a formal database compatible with Technical Architecture	Typically relational with ODBC or SQL 2 access. DRDA also acceptable. Object DB OQL, ODMG 2.0	Application Delivery Architecture, Technical Architecture, Platforms
3	Data Model is available and expressed in a standard form	Normally Entity Relationship Diagram or UML Class Diagram. Meta Data online	
4	Runtime environment is compatible with Technical Architecture	Relates to processor family, DBMS, operating system, etc.	Technical Architecture, Platforms, Application Programming Interfaces
5	Supplier is approved, stable and has sound strategy	Local support capability preferred	
6	Meets functional requirements		Business Architecture, Processes
7	Application will scale to anticipated volumes (5 years)	Check file (batch), transaction and number of users limits	Business Architecture, Products and Services

Figure 14 - Sample Selection Criteria (Application)

Specifying the To Be Architecture

In specifying the future architecture, we will chose options. Timeframes are also very relevant. We will identify elements that:

- we have now and wish to continue to promote
- we have now and want to discontinue at some point in the future
- we want to investigate and pilot with a view to adoption
- we want to phase in at a future time
- we have now and will continue to use, but which should not be more widely used

The detailing of the future architecture must distinguish these categories and link them to the relevant timeframes.

Overview of Architecture Implementation Process

Implementing an architecture driven approach in an enterprise is no small undertaking. It is often a culture change and fraught with all the difficulties this implies. Organizations are also under great competitive pressure and cannot afford to stop the mill to redesign. We have thus devised an approach that allows some major benefits to be achieved quickly, while providing for more benefits and greater rigour in the longer term. It is illustrated in the accompanying figure and includes some major elements.

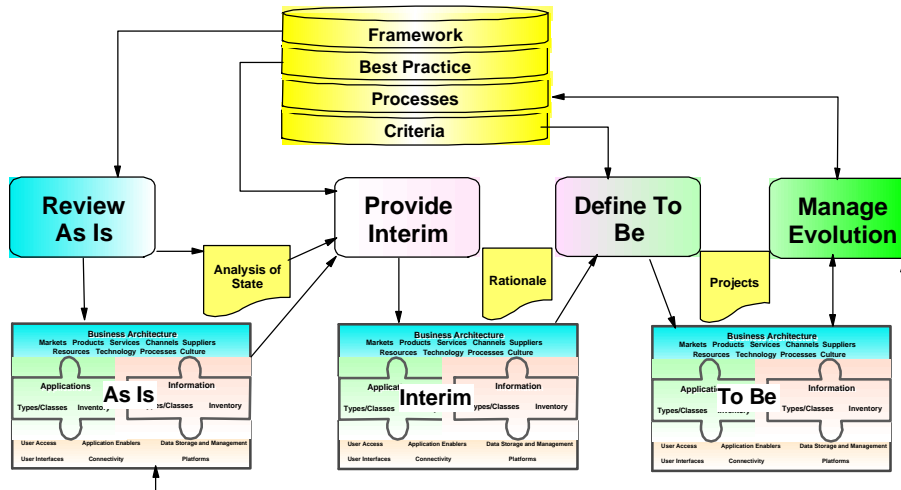


Figure 15 - Architecture Capability Process

- The **repository** indicates things that can be provided by the frameworks and consulting assistance. These include: the frameworks, best practice choices, process guidance and choice criteria
- The first step in implementation is usually **gathering and reviewing** the as is picture. With appropriate tools, this can be achieved on a distribute (geographical and responsibility) basis. A review of the as is picture will highlight serious gaps and opportunities.
- Next, we can provide **interim guidance** based upon the current architecture, industry best practice and the results of the analysis of “as is”. This allows projects and decisions to continue unabated, while preventing bad decisions
- As analysis proceeds, parts of the **“to be” picture** are fleshed out and initiatives will take responsibility for realising these via acquisition, construction or other activity. As these proceed, the “to be” gradually becomes “as is” and this picture can be updated.
- Ongoing assessment can determine how we are progressing and where we need to adjust to stay on course or **improve** the rate of progress.

Management of Architecture Artifacts in Archi

In documenting, analysing, designing and refining architectures, we will generate a great deal of knowledge. In many settings, this is managed in an ad hocacy of tools including word processing documents, presentation files, spreadsheets and end user databases. Some may be in corporate intranets and modeling tools. We did much the same for a few years, evolving a set of tools to assist us. Finally, we decided to develop a purpose built knowledge management tool to integrate the meta models, frameworks, collected data, future architecture plans and supporting documents and models. The result is a multi-user, web based, repository and knowledge management tool called Archi.

Archi provides concurrent support to architects, strategists, program managers and meta modelers as well as system administrators. It lives on a server, between a standard web server and relational database product. It manages a repository holding structured information and related documents which can be versioned. It can also reference external “unmanaged” content. Clients access the repository and tool functions through standard web browsers.

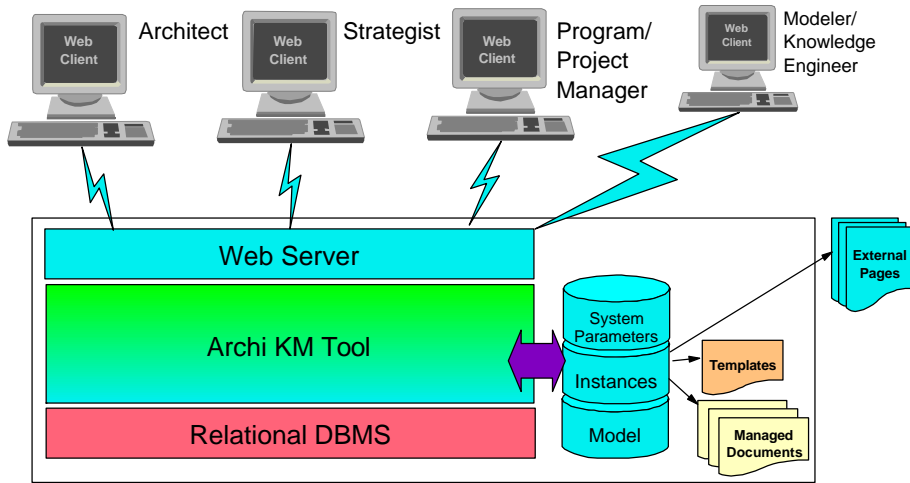


Figure 16 - Archi Architecture

The Inspired Frameworks© meta models have been fully implemented in the tool. Archi can thus support the collection, evolution and analysis of the full spectrum of architecture knowledge, as well as supporting documents. Documents can be of any type that the client workstations can handle, including Word™ documents, Powerpoint™ presentations, spreadsheets, models from CASE tools, project management files etc.

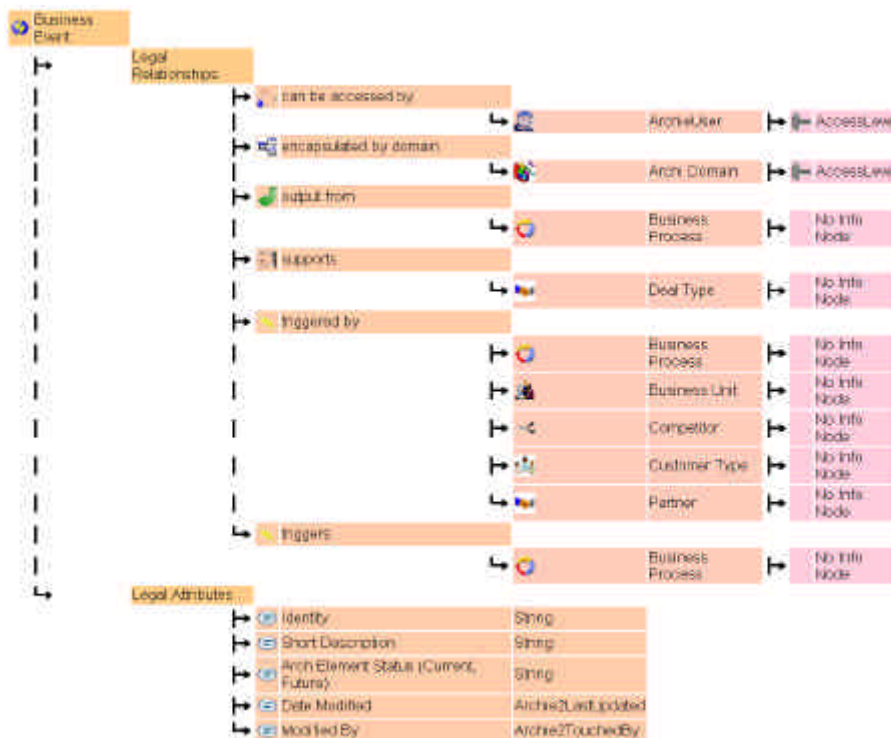


Figure 17 - Example of Architecture Type Definition in Archi

A variety of browsers provide different perspectives on the knowledge, including:

- Basic capture, editing, viewing, relating
- Cross referencing (matrix view), plus inferring relationships not defined directly

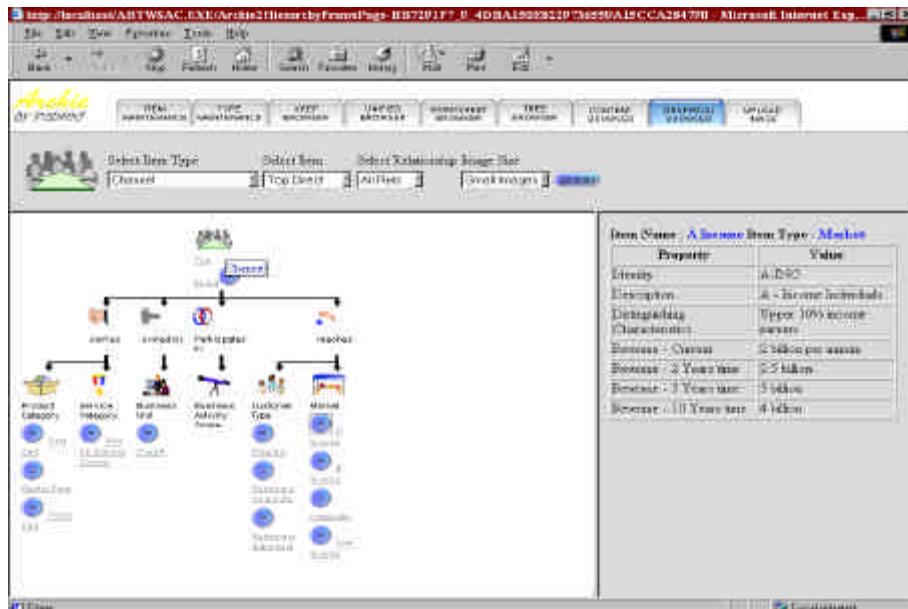
- Drilling down from spatial models of the frameworks to query underlying repository content
- Graphical navigation through the knowledge space
- Filtering by current, future scenario etc.
- Integration of content into other web sites via “live” queries
- Custom views, where the look and feel are fully specified by a developer

We have chosen to illustrate a couple of these views by including some screen Captures.



The **Spatial Browser** provides a way of viewing topographical, geographical or conceptual maps as an aid to locating relevant knowledge within a context. Drill down capabilities and queries on the knowledge base are supported. Unique custom “dashboards” can be created.

Figure 18 Spatial Model Interface to Architecture Repository



Items and relationships can be viewed graphically in the **Graphical Browser**, which allows querying items within the concept space and refocusing to any displayed item as the anchor point. The Graphical Browser can also follow single relationship types recursively - for example to display a hierarchical structure

Figure 19 - Navigating the Knowledge Base Graphically

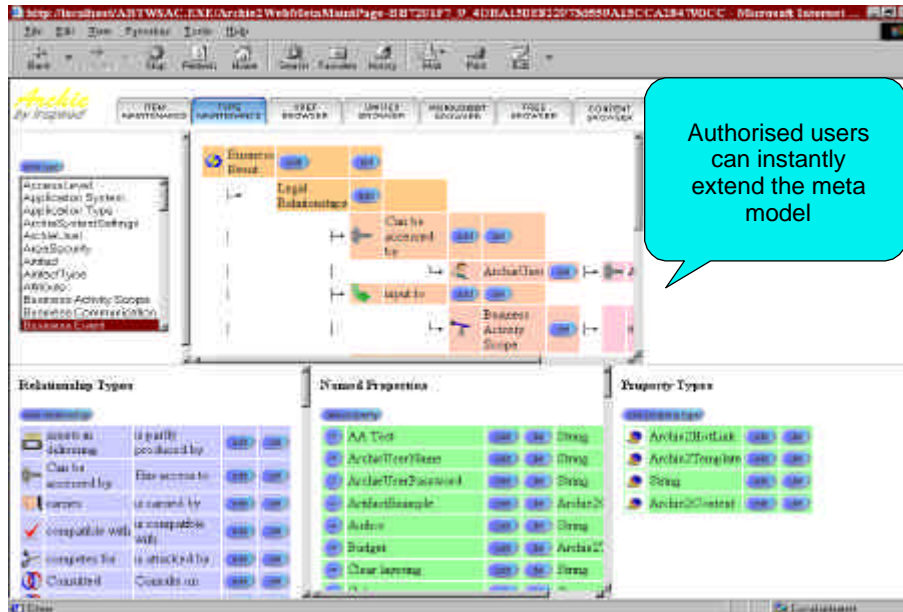


Figure 20 - Extending the Meta Model

Archi is very flexible. An authorised user can extend the meta model with new types, relationships and attributes via a provided web interface environment. These extensions are immediately usable by the creator, and require only security permission linking to be made available to other users. This can occur at run time, with concurrent use of the tool by other users, and no down time. The browsers and interface patterns will immediately adapt to provide full functionality for the added structures. A screen capture of the meta model editor is shown in the accompanying figure.

Security

Archi has a powerful security model that relates users to a profile that determines which functions within the tool may be used, what types of knowledge a given user can access, and what may be done (view, create, update, delete, import or export). Group privileges are supported and users may belong to multiple groups. Domains allow the grouping of types into logical (not necessarily disjoint) subsets relevant to different roles, simplifying security administration.

Filters

User defined filters allow control over what is viewed. These can be based upon values of properties of selected types as well as relationships. For example: Viewing everything linked to a particular business unit; viewing tasks which have not been allocated to any responsible party (hence no “allocated” link); or perhaps viewing information added within the last month only.

Templates

Archi supports the definition of default values and templates. For example, one could specify that a project should have a budget expressed as an Excel™ spreadsheet, a business case expressed as a Power Point™ presentation, a risk assessment which is a Word™ Document and a User Requirements Interview which is an .avi video file. We may have several other attributes, such as the name of the sponsor, the inception date of the project, current status etc.

Once the structure is defined to Archi and templates are provided for the relevant items, a user creating a project will automatically get a set of these templates cloned and associated with the new project. These can then be easily opened from within the Archi environment and customised to the project. Archi can version these artifacts for you as well, providing a secure knowledge base on a server. This kind of power can prevent the loss of valuable knowledge when staff move on, and can allow much greater leverage from existing assets, since they are easily findable and identifiable as to their currency, state and relationships.

Integration

An “anonymous mode” feature allows generation of queries into the Archi knowledge base. These can be embedded in web pages or other document types supporting URL’s and allow Archi content to be accessed without logging on and within existing web sites. This facility can be deployed very quickly - in one instance a production software fault tracking and release management system was conceived, modeled, implemented and live on the Internet supporting users on two continents in less than a day.

Customization

Customizable Menus. Authorised users can create customized menu's which are associated with a user or user group, thus allowing rapid access to frequently used functions and content.

Custom Views allow the rapid creation of completely customized user interfaces, using an API to the underlying functionality. These can take on the look and feel required by a user community or integration with a corporate web site or portal. Views can be added to the tool at run time without interrupting operation.

Collaboration

A “discussion” data type allows easy linking of threaded discussion groups (like news groups) to any knowledge item. This can facilitate work of teams collaborating in a given area.

Tool Integration

Archi can import and export both type definition information and instance data as XML. A component is available to allow other applications to query the repository over the web, subject to security.

Conclusion

The combination of the Inspired Frameworks, the architecture capability process and Archi tool support provides a powerful infrastructure for really doing enterprise architectures and making them “live” in corporate settings.

For more information, please consult our web site or the writer.

① www.inspired.org ✉ mcleod@iafrica.com

References

1. "A Framework for Information Systems Architecture." John A. Zachman. *IBM Systems Journal*, vol. 26, no. 3, 1987. *IBM Publication G321-5298*. 914-945-3836 or fax: 914-945-2018.

2. "Extending and Formalizing the Framework for Information Systems Architecture." J.F. Sowa and J. A. Zachman. *IBM Systems Journal*, vol. 31, no. 3, 1992. *IBM Publication G321-5488*. 1-800-879-2755.