

Enterprise Architecture

By: John Dick, Holly Simmons, Maureen Vavra, Steve Zoppi in conjunction with Harris Ken's Enterprise Computing Institute

John Dick, Holly Simmons, Maureen Vavra, Steve Zoppi provide the following anecdote to describe the CIO's architectural reality today...

Joe Marketing VP: *"Hi, Bob. Just wanted to let you know that our Sales and Marketing division has purchased the XYZ CRM solution to fulfill our strategic goals for this coming year. I wanted to set up some time with you to discuss how we get our software set up for our marketing and sales folks. Do you have some time tomorrow?"*

Bob CIO: *"I am concerned that I was not included in the decision to purchase this software. There are many things to consider with respect to our architecture and we need to understand what is required to implement and support the product. What are you planning to use this software for?"*

Joe Marketing VP: *"Oh, you know, CRM stuff. This is the best package on the market today. I'm sure you'll be able to get it working easily."*

Bob CIO: *"Joe, were requirements written up for this? Was a technical review of the software performed?"*

Joe Marketing VP: *"Requirements? We had several demos and decided that this product would best meet our needs. The XYZ salesperson can hook you up with some techie guys from their company. I'll get you the contact information."*

In this article the authors provide an introduction to the enterprise architecture and discuss:

- *The practical steps necessary to create an architecture strategy;*
- *How to avoid the pitfalls frequently associated with architectural planning and implementation;*
- *Why it is important to invest time up front in defining a successful architecture to avoid frustration and expense later in the process;*
- *Why architectures should not be developed as a pure product development engineer might; and*
- *Why it is important to focus the architecture on customers, users and the processes that enable their success.*

Are We Having Fun Yet?

Being a CIO in today's environment is a tough job. Vendors target marketing, sales, and customer support executives in companies to sell their enterprise solutions, frequently resulting in a sale without any involvement from the IT organization. In their attempt to keep the sales cycle short, vendors are happy to avoid the difficult, technical questions raised as part of a product evaluation. Unfortunately, the business users do not typically understand the potential architecture, implementation and support issues, and do not ask all of the pertinent questions, thus resulting in a decision based on only a fraction of the necessary information. What is most difficult is that the CIO is the one left holding the bag when this happens, taking responsibility for making the solution work. Vendors enjoy lauding their product as straightforward to implement, easy to integrate, and simple to support. They even make claims that their product solves your enterprise architecture problems for you, but the reality is that these products are simply a point solution with an enterprise architecture smoke screen. The CIO and the IT organization must expend a great deal of effort figuring out how to make it all work harmoniously together.

A CIO career is truly not for the faint of heart. Architectural challenges are at every corner with varying standards, technology, integrity, and scalability that can leave you with a hodge-podge of incompatible equipment, software, and data. But this is part of the challenge and opportunity for a CIO! You and your team have the opportunity to identify what works, what does not work, and there is always a great deal of change and variety in the technologies and available solutions. You are getting paid to play with new technology and solve interesting, although difficult, architecture problems. Appreciate the challenges and rewards ahead of you!

Overview

Our focus in this article is to provide you, the CIO or technology manager, with a “walking tour” of enterprise architecture. The intention is not to provide a detailed reference manual for implementing an enterprise architecture strategy, but instead to offer an introduction to enterprise architecture, a practical outline of the steps necessary to create an architecture strategy, and how to avoid the pitfalls frequently associated with architectural planning and implementation. Whenever possible, references to other sources of helpful information have been included.

What is Enterprise Architecture? We define “enterprise architecture” or “target architecture” *as the framework that encompasses your corporate data, network, infrastructure, security, and applications, providing a foundation to support the business needs, processes, and information.*

While traditional definitions of the term “architecture” focus on the design and building of structures, this same concept can be applied to enterprise architecture. When building a house, your foundation must be built with care since every other component of the house relies on it. Your enterprise architecture is your foundation requiring stability and flexibility only gained through careful planning and design. Similarly, a shoddy house foundation can result in a structure that is unstable, unable to be expanded or added on to, and in many cases, unfit for living. The same is true for poor enterprise architecture. Careful planning

that incorporates needs for future growth or changes allows your organization to avoid costly replacement or being left with an inflexible, inefficient, high maintenance foundation.

As a CIO, planning your architecture ensures future viability, a solid return on your technology investment, as well as job security. It allows you to minimize downtime, eliminate technical incompatibilities, and enforce smooth operations. It also assists you in controlling your staffing plan and costs.

When it comes to planning your technology architecture, the “pay now or pay later” principle applies. Invest your time up front in defining a successful architecture to avoid frustration and expense later in the process.

The Classic Architecture Approach

Most organizations use a formal method to do the initial development and do significant updates 2-3 times a year. The underlying persistent blueprints, standards and procedures must be assigned owners and updated as needed, and published widely so that internal resources as well as vendors are aware of them.

In an area such as architecture, it is essential to build a foundation of organizational understanding regarding the need for standards. This may be a real challenge in a shop that is heavily into departmental computing. To do this you may need to establish or re-verify principles for major internal IT processes – articulate why you need a formal architecture and do this in a group setting so that key technical leaders in your organization (and any others which impact your environment) buy in. Beyond that, policies – definition and decree of your key ones - are the true infrastructure for architecture and should represent what your organization believes is needed and is willing to live by and enforce. They are particularly critical in the data and security area, as they set the proper foundation for everything else.































In general, the classic approach requires that you start with an end state vision statement of what your organization wants to put into place for architecture. These frameworks can be “borrowed” from various sources, but should be tailored to you own needs. It’s all about explicitly stating that your future direction is a layered, component oriented model, based upon industry standards whenever possible, rules and API orientated, and as open as possible. You ultimately will be developing a Target Architecture based upon business need, assessing your current state against that target, and building a migration plan to get there.

Most classic approaches today rely on a formal system, such as the Zachman framework, Whitemarsh Knowledge Worker Framework, and various tools or methods. The idea is to create a common model at the business level, and cascade top-down down to specific layer definition and the accompanying standards to support these.

The Zachman Framework was created John Zachman in the late 1980’s; he was a senior engineer at IBM at the time and created this model, which has stood the test of time. It is intended to straightforwardly depict an Enterprise model for an organization. It shows the designs or documents that represent the intersection between the roles in the system design process, that is, *owner*, *designer* and *builder*, and the product definitions, that is, *what* (material) it

is made of, *how* (process) it works and *where* (geography or network) the components are relative to one another. It is a very useful logical tool to help understand the extent to which your Enterprise is made up of multiple subsystems that interconnect.

ENTERPRISE ARCHITECTURE - A FRAMEWORK™

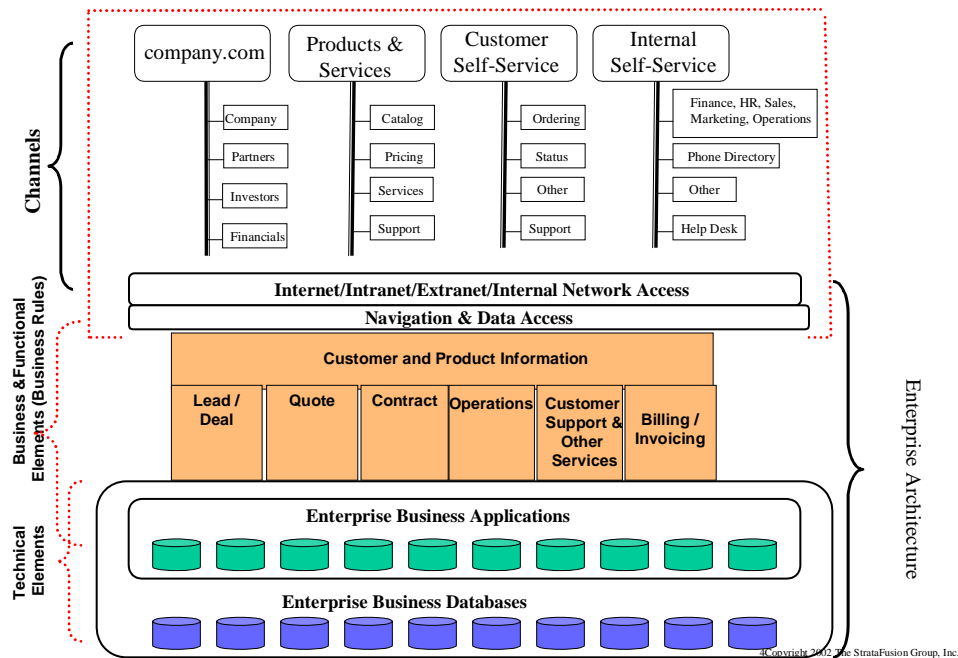
	DATA <i>What</i>	FUNCTION <i>How</i>	NETWORK <i>Where</i>	PEOPLE <i>Who</i>	TIME <i>When</i>	MOTIVATION <i>Why</i>	
SCOPE (CONTEXTUAL)	List of Things Important to the Business 	List of Processes the Business Performs 	List of Locations in which the Business Operates 	List of Organizations Important to the Business 	List of Events Significant to the Business 	List of Business Goals/Strat 	SCOPE (CONTEXTUAL)
<i>Planner</i>	FNTTY = Class of Business Thing	Function = Class of Business Process	Node = Major Business Location	People = Major Organizations	Time = Major Business Event	Ends/Mean=Major Bus. Goal/Critical Success Factor	<i>Planner</i>
ENTERPRISE MODEL (CONCEPTUAL)	e.g. Semantic Model 	e.g. Business Process Model 	e.g. Business Logistics System 	e.g. Work Flow Model 	e.g. Master Schedule 	e.g. Business Plan 	ENTERPRISE MODEL (CONCEPTUAL)
<i>Owner</i>	Ent = Business Entry Rein = Business Relationship	Proc = Business Process IO = Business Resources	Node = Business Location Link = Business Linkage	People = Organization Unit Work = Work Product	Time = Business Event Cycle = Business Cycle	End = Business Objective Means = Business Strategy	<i>Owner</i>
SYSTEM MODEL (LOGICAL)	e.g. Logical Data Model 	e.g. Application Architecture 	e.g. Distributed System Architecture 	e.g. Human Interface Architecture 	e.g. Processing Structure 	e.g. Business Rule Model 	SYSTEM MODEL (LOGICAL)
<i>Designer</i>	Ent = Data Entry Rein = Data Relationship	Proc = Application Function IO = User Views	Node = IS Function (Processor, Storage, etc) Link = Line Characteristics	People = Role Work = Deliverable	Time = System Event Cycle = Processing Cycle	End = Structural Assertion Means = Action Assertion	<i>Designer</i>
TECHNOLOGY MODEL (PHYSICAL)	e.g. Physical Data Model 	e.g. System Design 	e.g. Technology Architecture 	e.g. Presentation Architecture 	e.g. Control Structure 	e.g. Rule Design 	TECHNOLOGY MODEL (PHYSICAL)
<i>Builder</i>	Ent = Segment/Table/etc. Rein = Pointier/Key/etc.	Proc = Computer Function IO = Data Elements/Sets	Node = Hardware/System Software Link = Line Specifications	People = User Work = Screen Format	Time = Execute Cycle = Component Cycle	End = Condition Means = Action	<i>Builder</i>
DETAILED REPRESENTATIONS (OUT-OF-CONTEXT)	e.g. Data Definition 	e.g. Program 	e.g. Network Architecture 	e.g. Security Architecture 	e.g. Timing Definition 	e.g. Rule Specification 	DETAILED REPRESENTATIONS (OUT-OF-CONTEXT)
<i>Sub-Contractor</i>	Ent = Field Rein = Address	Proc = Language Stmt IO = Control Block	Node = Addresses Link = Protocols	People = Identity Work = Job	Time = Interrupt Cycle = Machine Cycle	End = Sub-condition Means = Step	<i>Sub-Contractor</i>
FUNCTIONING ENTERPRISE	e.g. DATA	e.g. FUNCTION	e.g. NETWORK	e.g. ORGANIZATION	e.g. SCHEDULE	e.g. STRATEGY	FUNCTIONING ENTERPRISE

John A. Zachman, Zachman International (810) 231-0531

Whatever models or methods you choose, your organization must have models, tools and supporting processes to view individual IT projects, efforts in an Enterprise wide manner to ensure integration of:

- ◆ Critical data;
- ◆ Key applications;
- ◆ Enforcement of corporate policies – security, controls, data privacy;
- ◆ Common service levels;
- ◆ Unified user/interface to facilitate intuitive use of systems; and
- ◆ Viable product development if that’s what you do.

Enterprise Architecture Overview



This diagram provides an overall view the relationships of an enterprise architecture. To consider an architecture apart from its business context is to miss the point of what you as the CIO are trying to accomplish. The goal is to focus your architecture on accomplishing a business objective. You are generally not developing this architecture as a pure product development engineer might, but are rather driven by an enterprise business need generated by a requirement to sell product through a business channel that includes sales, marketing, finance, customer service, operations and product delivery, and human resources functions. The actions of these groups will result in a revenue flow and profitability calculation for the company.

For that reason your architecture is a melding of business channel considerations, which generate a set of business rules that create a functional roadmap, by which the delivery and support functions of your enterprise provide value through products or services. These business rules then drive a set of physical technical elements, which actually operate the technology component of your company's product or services delivery, support, and accounting processes.

The channels portion is the key starting point of your architectural quest. Here is the recognition of what your company's actual delivery proposition to the market is. This section is not focused on internal functional organizations as a traditional approach might dictate, but rather focuses on the ways product or services are delivered and those transactions communicated to your customers and eventually your internal enterprise processes. The focus of your architecture therefore becomes not your finance, engineering, or

manufacturing organizations, but rather your customers and the processes that directly support their success.

Once your channels and general information delivery requirements are identified, the business rules or functional roadmap portion of your architecture can begin to take form. Many information technology organizations start with this phase by focusing on functional organizations not business channels, and are therefore distracted from the true value they can deliver by the “requirements” of a functional organization which are important and certainly relevant to that organization and possibly the enterprise, but may be business processes that do not truly support the channels they think they support, but rather add potentially administrative overhead to your enterprise. This overhead may not be truly part of the essential channel delivery function and therefore is probably not as valuable a target for technology investment or return on investment (ROI). By maintaining focus on your key channels you can determine key functional business rules and processes which deliver value to your customer, but yet still allow for proper accounting of their transactions with a minimal amount of overhead. Yes, this does mean that some functions may be better not assisted by expensive technology, but isn't that appropriate?

After you have identified your key channels and derived from that your key business rules and developed a functional roadmap, you are ready to start addressing the technical elements that constitute the heart of your enterprise architecture.

Necessarily this starts with your network infrastructure. In today's world the internet and associated web technologies create the window into your enterprise certainly for customers and even for internal users of your systems. Therefore, Internet, intranet, extranet, and browser-driven access to your information systems have more recently dominated the focus for your network infrastructure.

Also integral to creating this window is the architectural component that deals with the navigation and data access element of your architecture. This layer which is intimately connected with your fundamental network structure and feeds into your business rules and functional roadmap processes culminates into a controlled look, feel, and access into your applications and database architectural layers.

The final layers of your technical architecture are the heart of the business rules/logic and implementation of your functional roadmap, the applications and databases direct, store, and manage the specific functions of your business. Based on channels, business rules, and business processes these technologies whether custom developed or purchased vendor solutions form the heart of your systems. Integral to these layers, but architecturally separate are the enterprise application integration (EAI) components and inter-enterprise components, which provide the pathways that allow disparate applications and databases to interact enabling the channels and functional roadmap.

In the next few sections we will spend more time looking at each of these layers.

Planning for an Enterprise Architecture

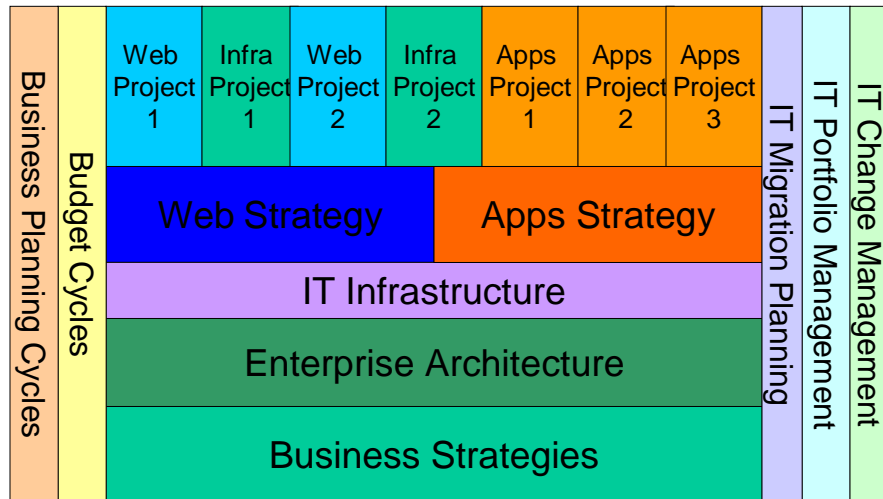
Enterprise architecture may not be the most obvious contribution provided by the IT organization, but it is integrated into many aspects of the business. As a CIO, you must be aware and able to plan for these interdependencies when budgeting, resourcing, calculating your return on investment, or even as part of your daily operations and internal processes.

Aligning IT with the Business

While IT is a support organization, IT also has a very strategic role in defining and driving technology decisions to support the business. This definition cannot be performed by IT in a vacuum, but instead should flow directly from the corporate goals or objectives.

Assuming the executive team is defining the direction and needs for the company, and each business unit is interpreting what that means to their organization in fulfilling the objectives, it is the CIO’s responsibility to plan for sustaining systems support, focus on strategic IT needs such as infrastructure growth or systems to improve efficiencies. In addition, the CIO drives the translation of the business needs into technology needs, providing a “check and balance” to the business units in an organization, helping to avoid investment in technology that may become “shelfware”, never implemented, or even worse, implemented, but not used. Also as part of this analysis process, IT must also define what architecture changes or additions are required. This is imperative in determining the current and future architecture plan as well as determining budget and staffing requirements.

IT Strategic Planning Process



MWV 7/12/02 V1.0

Aligning the architecture with business strategy and processes is essential. The obvious follow-up to this is that every major business change must initiate a re-examination of your IT architecture for impact, and the more formal the better.

The key is to ensure that all the vertical processes on the right side of the chart include architecture in some form or another. The Migration Planning should be a direct result of

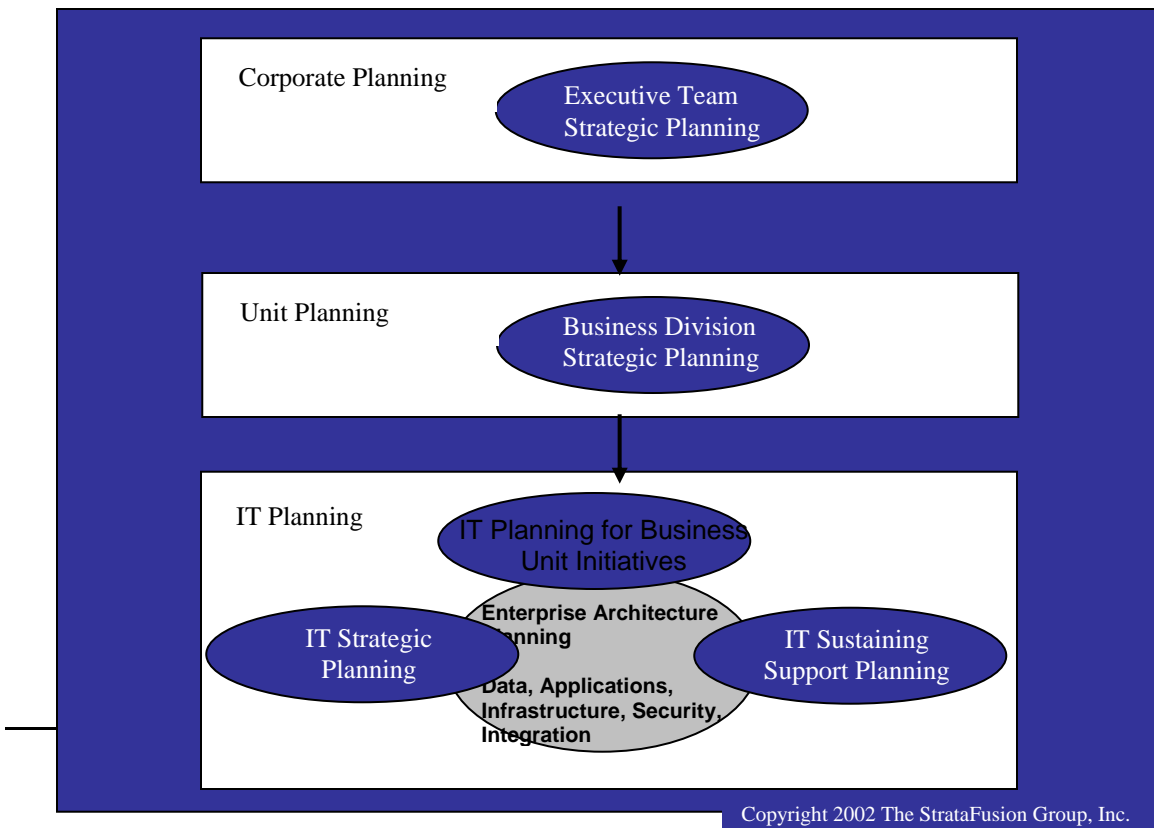
establishing your target architecture, the Project Portfolio should be managed by (and have a project methodology that) supports it, and the Change and Configuration Management processes must incorporate architectural planning at the logical and implementation level.

Ongoing, whatever architectural models and associated processes you adopt in IT, they will only have value if they keep that link to the business. To have integrity, these architecture models must themselves be standardized and integrated, enterprise wide, and *they must be observed by all new initiatives and projects if you want to sustain a strategic direction.*

This link is critical to keep in mind, yet we all forget or ignore it regularly. Here are some hints to help stay on course and mitigate damage if you don't:

- ◆ Don't make the initial architectures, policies, rules and processes so involved and difficult that no one will follow them; Link them to the business culture;
- ◆ Pay people to stay in touch with industry direction and incorporate that knowledge into your plans;
- ◆ Set up architectural reviews early in project life-cycles and allow course corrections when new information indicates they are appropriate;
- ◆ When you set up your architecture, follow industry models and standards whenever you can [and if you have to wait 6 months to do so – do it]; rolling your own and retrofit are hard work and make your direction tough to anticipate, and it's much harder to integrate packages;
- ◆ Set the "Target Architecture" goals for 6-18 months out so that new projects can anticipate what's coming; and
- ◆ Have a migration plan for non-compliant systems and use it as a "plan B" to retrofit the renegades that sneak by your standards and get implemented.

Another more organizational way of looking at it might be:



Alternate view of the IT Planning process including enterprise architecture planning.

Budgeting an Enterprise Architecture

Defining requirements or changes to your enterprise architecture should be an integral part of the IT/Business alignment process. Your major architecture costs are identified as well as specific cost drivers that determine the options available. The chosen options for your budget will depend on corporate growth plans, acceptable risk factors, and finding the best fit for your organization. The whole process can be daunting, but if you ensure that each initiative is analyzed, this will help to avoid excluding key items from your budget.

In addition to the architecture costs associated with the business unit plans, the IT group must also be analyzing their own strategic plans for costs that need to be included in the architecture budget. This process assumes that your IT organization has a current and future architecture strategy in place and that both sustaining and forward-looking initiatives will be addressed. Again, the options can be analyzed and specific costs associated with the budget.

Completing these steps will take you past the first hurdle. Compiling this information is not easy, but it is necessary in order to avoid what can frequently seem to be “hidden” costs.

The second hurdle in the enterprise architecture budgeting process is obtaining buy-in for the proposed budget. A well thought out budget plan where each cost is tied to a corporate initiative is easier to justify and can streamline the budget approval process. It needs to be clearly articulated that any strategic initiative with architecture requirements must be funded in full. In other words, the company cannot choose to back the business initiative without supporting the IT initiative behind it. Architecture expenditures can frequently be viewed as optional due to lack of understanding of the purpose. The information you collected in the first part of this process in addition to ongoing education will assist in communicating the importance of supporting the architecture budget.

Structuring an Organization to Support Your Enterprise Architecture

Defining, developing, and maintaining an enterprise architecture is a big job. In spite of this, many companies tend to neglect the importance of having employees who specialize in enterprise architecture. Frequently, CIOs rely on application developers who understand the intricacies of developing software, but do not have a firm grasp on the entire technology picture and the interdependencies between applications, database, network, security, and infrastructure. This is not to say that a developer is incapable of defining a successful architecture, but instead to stress that enterprise architects are senior contributors with a wide and varied background in technology. It is worthwhile for developing or maintaining a solid architecture to invest in architects who have the proper set of skills. An architect's role is not just based on technology knowledge, but also strategy and leadership since it frequently involves evangelizing a solution, not just defining the solution.

It may be challenging for a CIO to justify architects on his staff. When headcount is already limited or there is not a need for a full time architect, this role can be filled or supplemented by other internal staff or consultants. There is some amount of risk associated with this approach and the CIO must weigh the options. Relying on technical staff with a limited background may save budget in the short term, but being short sighted or not being able to analyze the potential architecture issues can lead to costly fixing or replacing later in the process. When bringing in consultants, it is important to focus on the level and breadth of their experience and it is frequently necessary to hire multiple consultants who are specialists in one particular area such as security. By not hiring at least one experienced architect, who perhaps can share other duties in the organization, a CIO may be putting his company's architecture at risk. What may appear to be a large cost savings up front by not having the headcount, can lead to a great deal of expense and frustration down the road.

Establishing an architecture group – even if only one person – is critical to creating a balanced IT organization. Similar to the reasons for separating Development and Quality Assurance, Architecture should be separate from Development or Infrastructure. It is not a successful approach to have your Development group testing final code before providing to the customer, and in the same way, having Development drive architecture design may be pulling them away from their core competency of application design and development. As shown in the figure below, your Architecture group should be integrated across your IT organization. The architect should be involved in the early stages of design with the Program Management group to focus on software or tool evaluation, high level information or application design, and involvement in driving standards, policies and procedures. Additionally, interaction with Development, QA, and Infrastructure occurs regularly to focus on integration, security issues, code reviews, and general research and development.



Copyright 2002 The StrataFusion Group

Depiction of a typical IT organizational structure that separates each entity to provide a system of “checks and balances”.

Architectural review and fit assessments for systems, technology, major changes

Once you have architectural models and standards, you can then assess all significant new proposed projects against them for fit. Most organizations do this at least twice in the project lifecycle; once at project initiation to determine the technology scope, fit and infrastructure requirements, and later to review more formalized logical designs prior to development and infrastructure build. Here are some tips:

- ◆ Make architectural review a pre-review a step for any IT project approval [watch for packages!]
- ◆ Have formal life-cycle(s) and/or methodologies with steps that require the production of standard documents that become part of your overall architectural model;
- ◆ Use an internal team of IT experts – it helps to develop people;
- ◆ Review against your standards and revise them if they don’t work;
- ◆ Reward compliance and innovation;
- ◆ Watch for “pilots” – they can become ingrained and hard to extract;
- ◆ Use the team to evaluate and make recommendations on the architecture migration plan;

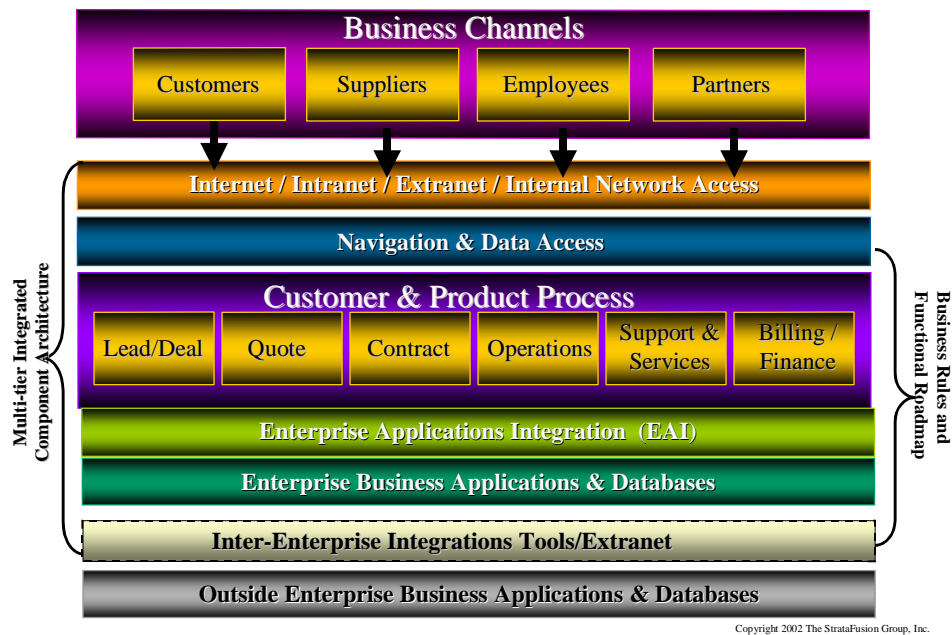
- ◆ Respond to changing needs by seeing non-compliant architectural requests as a signal of change. Several rejections of similar requests could mean you have a major gap in your model or infrastructure – use this to justify and fund upgrades;
- ◆ Communicate architectural review information as an early warning to the Change and Configuration Management processes; and
- ◆ Periodically use the team to evaluate and make recommendations on new technologies or business direction.

Change Management at the Meta and Operational Level – a Critical Success Factor

Change management, and the activities of Release and Configuration Management, are significant marks of maturity in an IT organization [ref SEI criteria for achieving Level 2 of their software capability maturity model.] Ad hoc and uncontrolled change in the environment threatens production service level agreements (SLAs), causes chaos in development and can wreak havoc with your architecture.

This is especially critical in the later phases of large projects where workarounds may show up as solutions to poor design. It is the operational, project-related decisions that critical people make on the fly which will truly enable you to sustain and advance a viable strategic IT direction. You best technical people can, just by trying to do their jobs properly, introduce disconnects which can completely derail your strategic direction.

However, you can use the Change Management process to protect your architectural investment. Proper release management review steps and environmental testing will identify non-compliant technology. The implementation of these processes effectively “puts a wall” around your production environment and forces changes to go through the proper life cycle, where architectural fit is assessed.



Enterprise Architecture and the relationship of a Functional Roadmap and Multi-tiered integrated component technical architecture

In this section we will introduce the concept of multi-tier integrated component architecture. This layered concept consists of:

- Network access (Internet/intranet/extranet/internal);
- Navigation and general data access;
- Customer, channel and product processes usually depicted as business rules and a functional roadmap;
- Enterprise application integration tools and interfaces;
- Enterprise business applications and databases;
- Inter-enterprise integration tools and interfaces; and
- Outside enterprise business applications and databases that you interface with at other companies.

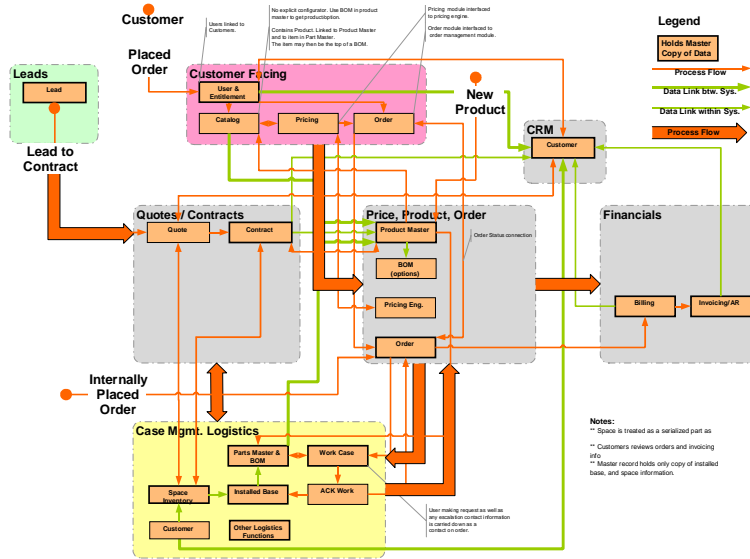
These layers form the primary units of your multi-tier integrated component architecture, and can be looked at in two parts 1) the functional architecture or roadmap and 2) the technical architecture. Each of these layers will be described in more detail in following sections. For now suffice it to say this concept can be depicted in many ways but the components remain much the same. The key point here is to understand that the functional architecture or roadmap and the technical architecture are distinct and different units which can be developed separately, but must be brought together in the overall architecture to

insure business synchronization. The key integrating layers in order of importance and impact are:

1. Enterprise Applications
2. Enterprise Databases (shown in the diagram above included with Enterprise Applications)
3. Enterprise Application Integration (EAI)
4. Inter-Enterprise Integration
5. Navigation and Data Access

The functional components of the architecture will determine the implementation of the technical components and therefore must be understood first. These rules must be uniquely identified then incorporated knowingly and carefully into the five areas above in the priority depicted. That means that the majority of these rules should be first of all contained in the Enterprise Applications layer. Once you have placed all those possible or technically feasible in that layer, you can then utilize the Enterprise Database layer for those business rules that are best kept there such as some data architecture design and modeling items, denormalization, data warehousing, stored procedures, or triggers clearly not appropriate to the applications layer. The EAI and Inter-Enterprise Integration layers will only contain those rules that govern the transportation of information from one application, database, or enterprise to another within the enterprise and should not include any functional business rules. The Navigation and Data Access layer should be limited only to those rules which impact data access and presentation to a customer such as what information will be displayed. Any attempt to actually process or modify information in the presentation layer must be avoided. Graphic delivery of the required information is all that should be addressed in this layer.

Functional Roadmap



Copyright 2002 The StrataFusion Group, Inc.

The Functional Roadmap or business rules are the most important part of your architecture. Without these definitions you will not know how to implement the technical portions. You will be making many implementation decisions during the technical portion, but ideally these will be more along the lines of configuration and data handling decisions as opposed to how the business is intended to be run. This does not lessen the importance of these decisions, but if you find yourself deciding how the business should operate in the technical architecture then a problem is rapidly developing.

A good Functional Roadmap has four dimensions. These include:

1. They are driven by the relationships between the business channels discussed earlier and actual business processes which deliver to those channels. They are best depicted graphically in flow chart manner as shown above, starting with the larger overall processes such as quotes, lead generation, or customer service, and then drilling down into each of those processes in enough detail to establish data points and information flow.
2. The major components of the roadmap are those that procedurally or functionally drive the business, such as order to cash or product master maintenance. These components are definitely not organizationally identified departments such as sales, operations or finance. Rather, they are the actual key processes driving the business channels.
3. The identification of the data points, units or types which identify the “what” of the information.

- The actual data links required and the specific data points which are linked or exchanged and identify the information flow, exchange or integration of the identified data.

It is also important at this point to follow some of the traditional forms of data modeling to make sure you are doing such things as collecting the data – particularly the master data – once and in one place, being careful not to duplicate data in multiple locations or collect it twice in two different ways or in the wrong process.

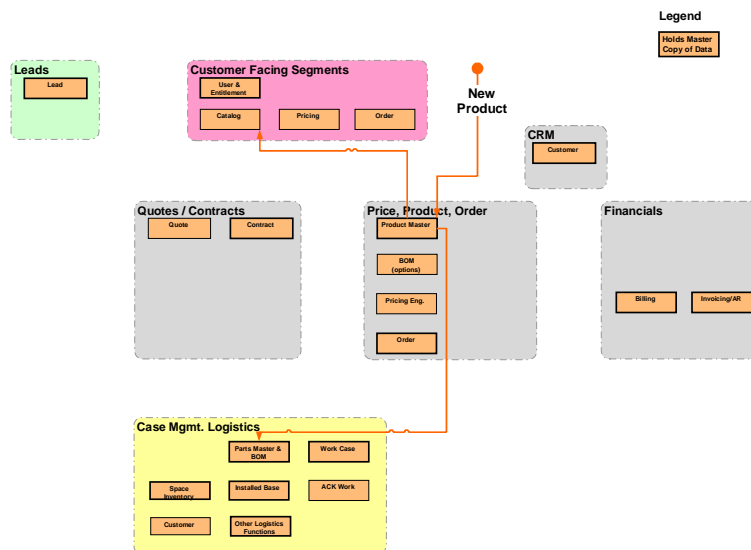
This drawing does not contain technical terms or tools identifying where functions occur such as in an ERP or CRM system, but rather focus on the business process itself identifying data points and flows to standard business processes not organizational departments.

Simply put it is the flowchart of your business starting with the customer identification and ending with the delivery and accounting of the value provided. It is the “what” and procedural “where” but not the technical “how”. That comes later.

The Functional Roadmap will usually be broken down into multiple maps associated with each general process of the overall map. These general processes may then be broken down into specific data points, data flows, and presentation which will be implemented by the technical toolsets applied. Without this crucial but often bypassed piece of the architecture the technical implementation phase is at best difficult and risky, and at worst futile in its chance for success. Unless you know what you are technically implementing or problem you are trying to solve you are most likely doomed to failure.

In the next section we will discuss several other examples of the functional roadmap.

Product Maintenance Data Process



Copyright 2002 The StrataFusion Group, Inc.

This section is an example of a “drill down” section of the previous overall functional roadmap to again illustrate what the next level might look like for a given process. This article is an introduction to some architectural ideas and unfortunately cannot complete your roadmap for you, but rather tries to show why you need one, what one might look like, and how it is critical to your overall enterprise architecture.

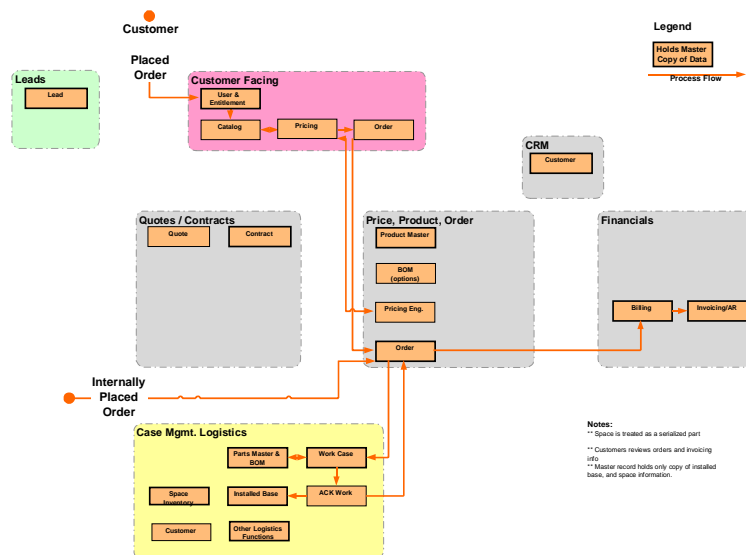
This example is the product master maintenance process, and reflects a real world process. Some key items to note are:

- The master locations for information are clearly identified;
- The required information flows are articulated; and
- The sub-functions are identified and their relationships to the other functions within the product master process are identified.

This sub-function can be clearly traced from its location on the overall diagram, which indicates its relationship and integration with the other major customer channel supporting functions.

This diagram can then be used to facilitate specific, most likely tabular, identification of required data points and flows which can be turned into a configuration, setup, or development specification for use by your technical architecture and eventual implementation plan.

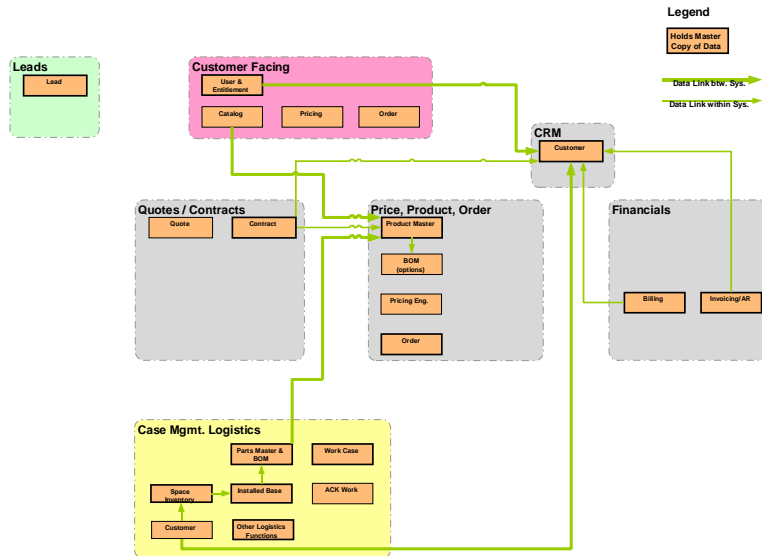
Order to Cash & Logistics Data Process



Copyright 2002 The StrataFusion Group, Inc.

Another example of a “drill down” section is the order to cash and logistics process again based on a real life example. This particular business function represents a solution for a company selling tangible products utilizing a ‘best of breed’ application approach. Key to the solution is pricing the tangible item relative to the logistical location, which drives the invoice creation and the collection of cash. Retrieving the data and processing it from the correct application must be designed carefully to ensure that the business user community does not enter in the same information in more than one-step or process.

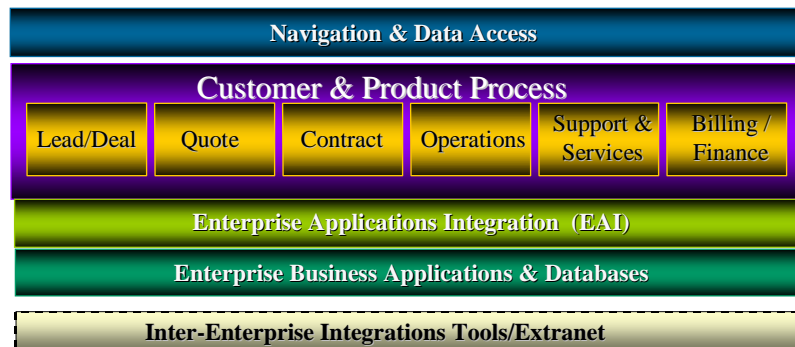
Data Integration Links



Copyright 2002 The StrataFusion Group, Inc.

This diagram again based on a real life example, although similar in appearance to the previous two, defines a unique and important component, that of the data integration links. This component is critical to your functional roadmap and must be included in all roadmaps. It specifically defines the flows and integration of your data points throughout the process(es), and while portions are incorporated in other diagrams this one should contain all of the required integration and be able to stand on its own showing the flows and integration of the required customer channel processes.

Business Rules and Functional Roadmap



Copyright 2002 The StrataFusion Group, Inc.

We have now completed our discussion of the Functional Roadmap and are proceeding to define and discuss the layers of the multi-tier integrated component architecture portion of your enterprise architecture. Before we do that though it is important to once again briefly emphasize the importance and relationship that exists between your technical architecture and your Functional Roadmap. These two together constitute the multi-tier integrated component architecture, which will deliver direct product value to your customer via the defined business channels and will also provide the processes and accounting necessary to control and operate your enterprise. These two major portions of your enterprise architecture are symbiotic in their relationship and must exist together. If you have one without the other then you have a technical infrastructure without clearly defined business rules, logic, and technology processes usually resulting in increased overhead, maintenance, and therefore cost efficiency and most likely effective customer channel delivery, or you have a good functional roadmap with a flawed or inappropriate technical infrastructure again impacting costs and effective customer channel delivery.

The business rules and functional roadmap provide the logic and flow pattern for the technical portion of the architecture which constitutes the nerve pathways resulting in the delivery of information and products to your customer channels. The total neural pattern and flow constitute your multi-tiered integrated component enterprise architecture.

In the following sections we will diverge into a series of discussions around the technology layers of the architecture and their inter-relations to the whole. These sections are not designed to constitute a technical treatise of each layer but to rather discuss them mostly in relation to the whole while demonstrating the importance and uniqueness of each major technological layer.

Network Access Layer

Internet / Intranet / Internal Network Access

Copyright 2002 The StrataFusion Group, Inc.

We will start our excursion into the technical portion of the multi-tiered component architecture with the network access layer. This layer is often most associated with those technologies which create the internet, intranet, extranet, and general internal access networks of your enterprise. Some of the distinguishing characteristics of this section include:

- Physical data communications access into and out of the enterprise such as DS3, T1, VPN, and co-location resources;
- Firewalls and DMZ's which define and defend the perimeters of the enterprise usually a combination of software and server/network hardware tools;
- Routers, switches, and other network devices, which provide intelligent routing of information in and out of the enterprise;
- Network cables, network interface cards and wireless access units which create the physical topology of your network;
- The browsers and other software data access components which actually control the presentation and physical access logic of your desktop and portable computer units;
- The transparent security of this layer via addressing schemes, port ids, and operating system roles and responsibilities are more transparently applied then in their layers; and
- Lastly although not addressed in any detail in this article, the computer units themselves which provide the personal tools to interact with and obtain and retain information from the enterprise architecture at a personal level.

In today's world, especially with the advent of the Internet's web-based technologies, the network is the glue that holds all the other layers together. There are essentially no business

rules involved in this layer, nor should there be. When operating properly it should be transparent to the using community, but without it each unit of the enterprise architecture becomes a “island of independent computing” and most likely little to nothing works at all.

This is the first layer that you encounter in any enterprise, but as we have explained it is also the most technically self-sufficient layer. Business rules and functional roadmaps have minimal impact on this layer, which is deeply immersed in the more pure technologies of TCP/IP, ports, topology, network addresses, switches and firewalls.

Some key considerations when thinking about this layer include:

- No business rules should be included here. This is the most purely technical portion of your multi-tier component architecture.
- A significant portion of your front line security is included in this layer, and its capabilities should be coordinated with the security capabilities of your Navigation and Data Access layer of your overall system security plan.
- This layer is often forgotten in the planning surrounding your enterprise architecture. Remember this is the physical doorway in and out of your architecture if the door does not open when appropriate or is too small you have no architecture only a wall or inadequate funnel.
- Make sure as you consider your technical designs you account for the capabilities and limitations of this layer. Some critical consideration include volume of data transported, geographical distances transited particularly for distributed database access or update, inadequate telecommunications bandwidth and/or bottlenecks, and single points of failure.
- Efficient database modeling and access schemes that minimize long seek times and large complex searches.

We will not spend a great deal of time on this layer, other than to emphasize its importance, since while critical to architectural success and the “beating heart” of information communication it pretty much stands by itself in the multi-tiered integrated component architecture.

Distributed Data Access Layer



Copyright 2002 The StrataFusion Group, Inc.

The distributed data access layer could be viewed as an integrated portion of the network layer, but for the purpose of understanding the multi-component technical architecture we will look at it as a separate layer. The reason for this is that it provides the presentation and common user interface (UI) which enables navigation, entitlement, and aspects of globalization, security, workflow and persistence which are impacted by certain portions of the business channels, rules, and functional roadmap.

This layer, especially when looked at from the Internet web or browser interface perspective, controls the look and feel of what a user will see, the initial navigation whereby he will choose and move among application toolsets, databases, and the presentation of data from these toolsets. Here the more visible security of passwords and “sign-on” methodologies is applied and requests for transactions or information are initiated. Some business rules are applied at this stage mostly related to the above functions of navigation and entitlement, these however should be limited to just these functions. The more complex rules of transaction, data integrity, data integration, and reporting must be assigned and maintained within the other layers such as enterprise applications and databases not at this layer. It is important that this layer utilize that information for the fundamental presentation, navigation, and entitlement functions, but the process of business rule structuring and adherence should occur within those tools and technologies. A mistake is very often made in this layer particularly by predominately Internet-based systems to incorporate and integrate these rules into the navigation and data access layer. This will almost universally result in a vast and complex duplication of capability and design, which is more appropriately inherent in the application, database, and integration layers of the architecture. This produces systems that are not only difficult and expensive to maintain, but also serve the enterprise poorly. One of the keys to success, and most importantly flexibility, in enterprise architecture is to be successful with this layer. If you develop your navigation and data access interfaces

properly then future incorporation of application and database changes become much easier. If you embed or duplicate functional and business rules within this layer you will end up with considerable duplicate effort and as a result significant complexity and cost into reacting to business condition changes. If you do not carefully utilize Change Control or document the locations of such duplications then you could very well cause your systems to be non-functional with little data integrity.

With the advent of Internet/web technologies such as the browser and a large assortment of web server, application, and emerging web services tools, many of the traditional limitations surrounding this layer have disappeared and been replaced with almost too many alternatives. The new power of this layer which started with the client-server age and came to near full fruition with the advent of the web, make this layer a true ally to the CIO's difficult task of common user access, standardization, and information integration. Revolutionary tools exist today to perform these functions, which were unheard of just a few years ago. This layer with today's technologies provides one of the most powerful advantages for systems in the last five to ten years. Admittedly it presents some new complexities and a learning challenge for technology professionals, but one can hardly imagine a world where such tools were no longer available.

Navigation and data access, when coupled with the next layer enterprise applications integration (EAI) sometimes referred to as "middleware," truly provide today's enterprise with a powerful one-two punch to the enterprise integration challenge. And, when you add some of the inter-enterprise integration techniques discussed below, which are starting to be provided, the CIO now has some truly significant tools in his battle to provide solid business systems information to his enterprise.

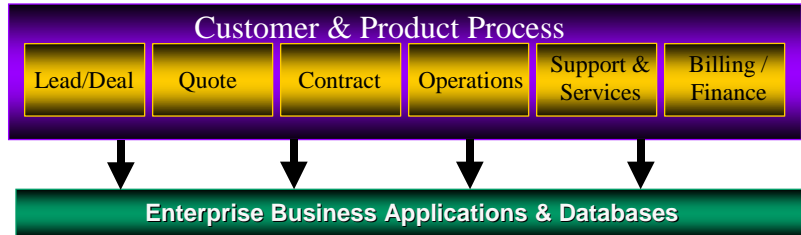
Initially small enterprises generally resolve their EAI issues with point-to-point, application directly to application integration. Once an enterprise has several integrating enterprise applications such as ERP, CRM, or multiples of each, it will start to find that point-to-point solutions are no longer sufficient or are becoming to complex for application interfaces. When an enterprise begins to require multiple interfaces among several similar or related applications it then becomes necessary to consider purchasing, or under some unique circumstances developing, a more robust and standard EAI layer. Many of the standard market EAI tools can be purchased, and depending on the requirements of your enterprise architecture, especially if more complex, can reliably be used in a somewhat standard fashion to integrate your enterprise applications and databases. These tools can however be expensive and are often complicated to implement due to what really becomes a more custom implementation and therefore more implementation cost intensive than originally perceived. It is important that every effort be made to utilize standard implementations of these products in order to keep configuration costs and complexity down to minimize follow-on maintenance costs. An alternative as long as your EAI requirements are relatively simple is to as we have previously said utilize point to point solutions initially or in a combination with carefully architected and thought out use of your navigation and data access layer to facilitate the sharing and inter-functional use of enterprise application information. In some cases the newer business process management and emerging web services tools may also be of assistance. This sophisticated approach to combining point-to-point solutions with solid and flexible navigation and data access layer design can hold off

the potentially more expensive EAI tool purchase and implementation for a considerable period of time. One must however be careful in the mindful, flexible, and well documented design of the navigation and data access layers and also be careful to plan thoughtfully ahead for the potential implementation of a more standard EAI toolset before your architecture becomes too unwieldy significantly complicating or decreasing the time available to implement a more standard, purchased tool approach.

The development of the architecture of this layer of your multi-tiered integrated component architecture is probably the most important for the growth of your enterprise as it reaches its more complex growth periods as a larger company. Be very careful to consider your strategy for this layer well before you need it and to proceed carefully and thoughtfully down this path. Although generally transparent to your user community this is a key area that can successfully simplify and standardize your EAI issues or conversely can make your enterprise architecture very expensive, overly complex and less dependable than today's powerful tools are capable of delivering.

The strides technology has achieved in the last five years in EAI technology is nothing short of amazing and can be of significant benefit to you as a CIO, and to your enterprise architecture.

Applications & Database Layer



Copyright 2002 The StrataFusion Group, Inc.

We are now going to discuss the functional and business heart of your multi-tiered architecture: the enterprise Business Applications and Database layers. These two layers, by design, should provide the majority of your business rules. We have noted a couple of potential exceptions to this in the implementation of your navigation and data access layer and your EAI layer, but to re-emphasize, these should be the exception rather than the rule.

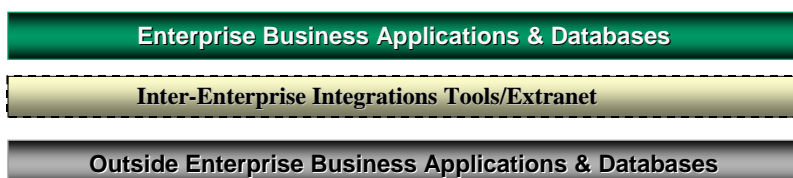
The Business Applications layer contains almost all of the design encompassed in your functional roadmap and associated business rules. In most enterprises this layer will consist of mostly purchased applications such as ERP systems like SAP or Oracle Applications and CRM systems such as Siebel, and your business intelligence and knowledge management systems such as SAS, Business Objects, Cognos or in-house developed reporting tools. In some cases internally developed applications that support unique business functionality will also be included. Extraction, Transformation and Loading (ETL) tools, which will also contain business rules around the aggregation of data, reside in this layer.

The functional roadmap and business rule requirements are and should be best contained within this layer since the applications themselves derive their capabilities from the configurations and implementation approaches driven by the functional roadmap. These applications then become the heart of your business process environment and channel processes. Here the traditional business functions such as finance, operations, customer service, sales, and marketing derive their basic functionality from the business channel driven functional roadmap. Here is the first time that the channel driven business rules and functional roadmap intersect with the traditional business organizations that should drive the configuration of applications to match not their structure, but rather the processes they support.

The database layer will usually contain almost all of the rest of your Functional Roadmap and business rules technical implementation, but only as it supports the Applications layer through such mechanisms as stored procedures, database triggers and the fundamental capabilities of a relational data model. In addition, your data warehouses reside in this layer. The key to success in this layer is a solid, well thought out relational data model for your applications based on sound relational concepts supported by good table design, careful key selection, and effective indexing. Through these mechanisms supported by effective stored procedures and database triggers where appropriate, you provide solid support to your applications layer where the majority of your business rules reside. It is emphasized that the only business rules that should be included in your database layer are those that directly support and are tightly coupled to the application through significant added functionality or effective design. Again business rules should be limited in this layer, but effective data modeling techniques should be used.

This layer also incorporates the data marts and data warehouses that support the analytical, business intelligence and knowledge management portions of your enterprise architecture. These databases normally in a de-normalized state and updated sometimes in mass on a regular basis have their own rules usually uniquely identified with each warehouse via a data model supporting its specialized capability. In this sense each of these constructs must have business rules associated with them, but since by definition they are reorganized duplicates of already maintained data the real key here is to have a well documented data model that is based on channel-driven business rules and identified in the functional roadmap. In that sense they are not truly a duplicate set of rules but rather a supplementary set that is appropriate to this layer. It should probably be noted here that the concept of an enterprise data warehouse has been discussed over the years. It is an unusual organization that can support and sustain such a large effort across even a small enterprise. Although data warehousing and its attributes are beyond the scope of this article and rather a subject of several texts, it has been my experience that starting with focused data warehouses or data marts on a smaller more distributed scale can provide focused value quicker than trying to create a single large enterprise warehouse.

Inter-Enterprise Integration



Copyright 2002 The StrataFusion Group, Inc.

The last layer we will discuss in the technical portion of the article is that of the Inter-Enterprise Integration tools layer. This is the layer that provides the organized access in and out of your enterprise for other enterprises. In this way it is similar to the Navigation and Data Access layers previously discussed, but directly targeted on the exchange of information between your enterprise and another one. In the context of the Internet and web infrastructure today, this is the home of your extranet and associated tools.

Some of the key ideas around this layer are to:

- Ensure it is physically but not logically insulated from your other layers;
- Ensure your security strategy isolates and controls access from this layer to your internal systems;
- Utilize standard interfaces to or from this layer using standardized approaches such as XML or EDI transfer or an in-house developed data access transfer layer similar to the one at the UI associated layers already discussed.
- Ensure these transfers are carefully thought out, standardized to preclude duplication of access methods and very well documented and under change control;
- Less is always better than more;
- Use Flexible systems thinking because these interfaces will be changing and will most likely need to support multiple enterprises most likely with different requirements; and
- This layer will support its own set of interface business rules and data integration roadmap, but should be a subset of, or dependent on, the internal business rules and associated functional roadmap that would constitute the master set. You will definitely want to have a clear line of demarcation between your internal systems and

these internal feeds with clear separation and precise transaction tracking incorporated as basic concepts in their development.

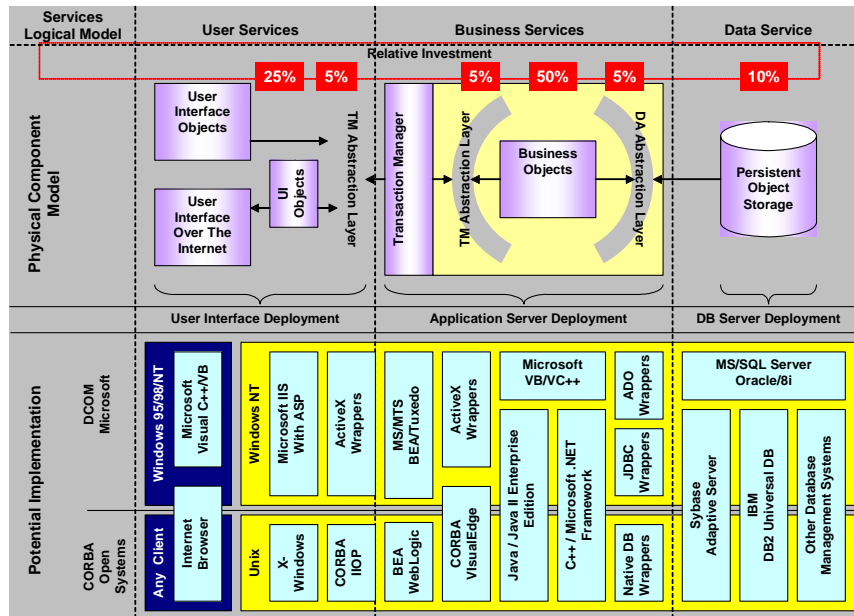
From an architectural standpoint this is clearly a tool layer. Any applications operating in this layer must focus only on the extraction, data exchange transformation, and actual transmission of inter-enterprise data and on applying only well documented business rules directly related to only those processes. Aside from transaction logging and associated functions database storage and subsequent enterprise use of information from this layer should be discouraged.

Developing a Strategic IT Portfolio

As mentioned earlier, IT must develop its own strategy, not only partnering with the business units to support their initiatives, but also determining what projects are necessary to support growth, improve efficiencies, and in some cases, to move the company to the status of a market leader or technology innovator. Review and assessment of the current architecture, both from a functional and technical standpoint, is key to defining an architecture roadmap and formulating a solid technology portfolio from which to operate and extend.

After understanding and developing a functional roadmap and multi-tiered integrated component architecture, the final step is to develop a technology roadmap and portfolio, which identifies the specific technical tools you will utilize to deliver ongoing solutions for your business channel requirements. This creates your blueprint and actual strategic portfolio of technology tools that will run your enterprise.

Technology Roadmap and Portfolio



At the foundation of any effective architectural blueprint is a well-rounded technology roadmap and portfolio. Shown is a graphical model describing a typical range of services required for a global client-server architecture. The model is not intended to be exhaustive nor definitive but must be significantly robust as to address the majority of interoperability, development and deployment needs faced by the organization. Although this model is predicated on a three-tier architecture, it can (and should) be extended into a full services-based architectural roadmap including an overlay of the four basic operational services: Transaction Management, Messaging, Directory Services, and Security Services.

At the 30,000-foot level the creation of this chart can be leveraged in multiple ways. Providing Application Developers, Architects and Management a consistent view of the application architecture becomes an extraordinary advantage when introducing new applications into the portfolio. The ability to organize the incoming applications' features into their logical strata can also give light to potential incompatibilities or highlight rough spots in application roll out and integration. For the implementation of units of work or functionality (as is done with commonly accepted Object-Oriented design principles), the chart can facilitate a walk-through of the target environment.

At the 10,000-foot level, when this chart is fully-completed and documented, it helps the Program Management Office, Engineering Groups, Architects and coders understand the interrelationships of the products they are producing and the frameworks to which they must fit. It is also extremely helpful as a tool to manage the reassessment and attrition of the "tools repository" which tend to become littered with unused parts over time.

At the 5,000-foot level, the chart is of significant value when starting the process of specifying projects of any kind. Because the architectural blueprint can be used as a checklist guide to the overall implementation of new technology or iteration on existing technology, there is less likelihood of an architectural oversight due to a lack of understanding of the existing environment. While this may seem unlikely, it is not uncommon that organizations of all sizes encounter significant oversights when performing preliminary application design only to discover that a critical component was overlooked in the infrastructure portfolio. For example, it is very common to overlook the value of "directory" or "messaging" services when architecting applications. The tendency is to take the simplest design approach of "letting the database do it all" but in reality, this is a very poor substitute for a scalable architecture.

There are other helpful reference elements in this form of architectural blueprint. The computations of Relative Investment (represented at the top of the chart) have been derived as an average from approximately 300 software engineering organizations but these numbers (or their projected values) can change drastically depending on the specifics of the project. Also of note are many elements seen in the "Potential Implementation" which can cross various boundaries. Note that Microsoft's tools are seen in the User Services and Business Services layers whereas the Sun (Java) tools are largely relegated to the Business Services tiers. The increasing competition between the two foundation framework players (Microsoft .NET and Sun Java in this example) makes it extremely difficult for an architect to maintain purity in the final implementation. The .NET framework has potential as of the time of this writing to limit Java's significance in physical delivery models whose implementations use

Microsoft operating systems in the application concentrator (Business Services) tiers as well as the client delivery tiers where browser-only clients are not viable.

The identification of a technology roadmap and portfolio which flows from a functional roadmap and multi-tiered integrated component architecture all derived from a sound understanding of business channel process requirements provide the technically astute CIO with a powerful and sustainable platform which will significantly contribute to his company's productivity and profitability, not to mention his own career.