

Nigel Martin^{*}
Shirley Gregor^{**}
Dennis Hart^{***}

CYCLIC ALIGNMENT AND ELECTRONIC COMMERCE SYSTEMS: THE ROLE OF ENTERPRISE ARCHITECTURES

Agile e-commerce systems are required in a world in which the environment and corporate strategy change rapidly. However structured formal approaches such as those provided by enterprise architectures (EA) are still required for large, global systems. It is suggested that the degree to which an EA framework allows for a cyclic alignment process should be assessed. This process involves both integration and feedback/exchange mechanisms among the component parts of an EA. A review of several proprietary EA frameworks shows all frameworks except for the Zachman architecture offer support for an integrating structure. Most link their constituent models with feedback and exchange channels, though some channels are unidirectional. None of the selected architectures, except for ARIS, have a discernible feedback loop to corporate requirements specification.

Introduction

Systems engineering processes that incorporate Enterprise Architecture (EA) concepts are sufficiently well understood for logical and physical systems to be designed and constructed. The enterprise architecture in its most simple form is a logical structure for classifying and organising the descriptive representations of an enterprise that are 'significant' to the management functions of the enterprise (that is, planning, leading, organising and controlling)

^{*} School of Business and Information Management, Australian National University, Canberra ACT 0200, Australia, E-mail: Nigel.Martin@defence.gov.au

^{**} School of Business and Information..., E-mail: Shirley.Gregor@anu.edu.au (corresponding author)

^{***} School of Business and Information..., E-mail: Dennis.Hart@anu.edu.au

as well as to the development of the enterprise's information systems (Zachman, 1996). There is evidence to suggest that enterprise architectures exhibit a strong practical application in commercial and government entities (Zachman, 1996).

Unfortunately, enterprise management and systems engineering processes, by their nature, tend to be protracted, with longer and more complex development cycles than might be considered optimal in fast moving e-commerce environments. For example, some recent views on e-commerce development methodologies (Pres-Heje and Baskerville, 2001) show tendencies towards ambiguous and fluid user requirements specification, widespread prototyping, frequent system releases and parallel development processes; all characteristics that do not fit well with the way in which enterprise architectures typically develop and evolve.

An example from industry illustrates the motivation for the current study. A large organization decided to implement an enterprise-wide system after management adopted a new corporate e-commerce strategy – virtualization of the organization (Wassenaar, Gregor and Swagerman, 2002). A formal plan was produced, a well-known ERP vendor selected, a budget agreed and a firm of consultants engaged to oversee implementation. From this point on the details of an EA were defined. Two years into the project, organizational management were told that it was not possible to complete the project without investment of additional significant funds. At this point a member of the senior management group admitted that they had very little choice but to allocate the extra funds – they no longer had any real understanding of what was being done by the project team. Moreover, in the intervening two years the organizational environment had changed and variations to organizational strategies were required. The project team, however, were completely occupied with the original project specification, and were mostly unaware of any change in requirements. Proponents of some EA frameworks would argue that an EA in this situation would serve as a tool for some common understanding, allowing two-way communication between system architects/builders and managements strategists. In this case, the EA did not fulfil this role.

Our deduction is that an EA framework should allow for what we term a *cyclic alignment process*. Any system constructed should be aligned with corporate goals. In addition, this alignment must be regarded as a *cyclic* process in the e-commerce environment. Management must have sufficient understanding on an ongoing basis of what is being implemented to ensure that corporate strategies are being attained, and project staff must have an ongoing understanding of any changes in corporate strategy.

Thus, our aim in this paper is to examine how enterprise architecture frameworks can assist with the rapid evolution of e-commerce systems that are aligned with corporate objectives. That is, how can an enterprise architecture support cyclic alignment? Our study is significant as this aspect of EA appears to have received relatively little attention.

The paper proceeds by first outlining the sparse literature that relates to the evaluation of EA frameworks and the concept of cyclic alignment. A number of proprietary EA frameworks are then evaluated in terms of attributes that can support cyclic alignment. The paper concludes with lessons learned from this evaluation of EA frameworks.

Conceptual background

Evaluation of EA frameworks and cyclic alignment

An enterprise architecture for a particular organization will incorporate a number of different modelling tools or representations of different aspects of the organization combined in a structured manner. The different representations or tools incorporated, and the manner in which they are structured, will depend on the particular EA framework adopted by the organization. Examples of proprietary frameworks include the Zachman EA framework (Zachman, 1996), CIM-OSA (Kosanke and Vlietstra, 1989), ARIS (Nagel, 2001), and the Meta Group EA (Westbrock, 1999).

We can find no previous work that proposes an overall set of criteria for evaluation of EA frameworks in a systematic way. Criteria that have been used in the evaluation of other, lower-level, modelling tools, include ontological completeness and clarity (Wand and Weber, 1995), support of principles of good decomposition (Weber, 1997), and simplicity and understandability (Navathe, 1992).

With respect to EA frameworks, Croteau et al. (2001) suggest "good enterprise alignability" as an additional criterion. The work of these authors suggests that the EA construct must be capable of facilitating enterprise alignment within the user context. Users must be able to use the EA to co-align business and information system objectives. This is achieved through the interactive consideration and joint development of enterprise strategy and systems, allowing both to be optimally matched.

A related concept is that of "enterprise integration" (Weston, 1999; Vernadat, 2000).

Weston (1999) concluded that it has become possible for an enterprise to capture, develop and apply formal models of itself, and over successive time periods use these models to decide how it might respond to new opportunities and threats. This Integration Structure is then applied to the various real (e.g., operational or resources) and virtual (e.g., business or process) components of the enterprise to enable change. Weston's research (1999) points to system components that will be small- or large-grained depending on the function. In order to facilitate change and reuse, the components will have an integration capability that will be capable of registering and accessing integration services from the 'Integration Infrastructure'. The Integration Structure and Infrastructure are highly complementary and equally required for system evolution. Figure 1 shows the Integration Structure and Infrastructure concepts.

Weston (1999) demonstrated that the application of an Integration Structure to Enterprise Models enables rapid system design, reconfiguration and on-going system development. The research shows that the result of a model-driven, component based approach to systems engineering is that the resultant system will have an inherent capability for radical or wide sweeping change. The embedded 'change ethic' should yield resource efficiencies, subject to well defined system decomposition and component design.

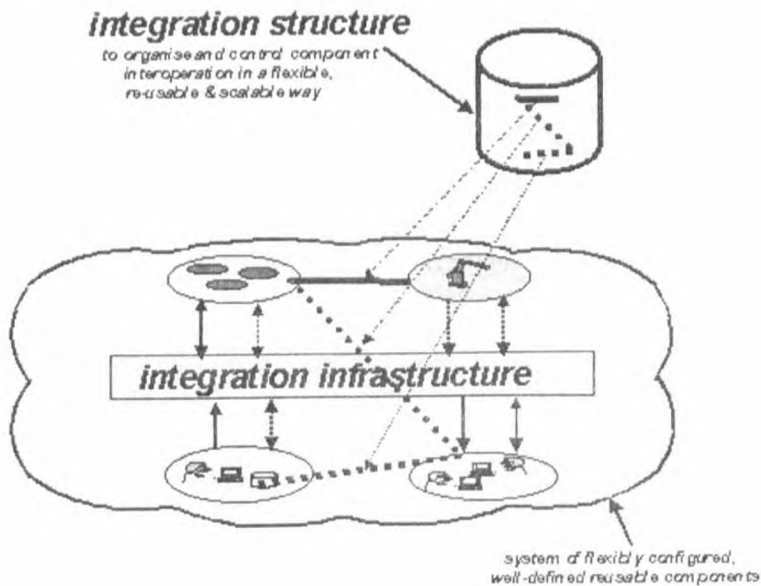


Figure 1 Integration Structure and Infrastructure (adapted from Weston, 1999)

Vernadat (2000) supports the use of an Integration Platform or Structure and the associated Infrastructure. Vernadat (2000) also actively promotes the use of reconfigurable, distributed agent-based architectures or models. This usage involves each model behaviour being implemented as an agent, and each model designed for easy modification or expansion without recompilation of the whole model.

From the above we can see that EA frameworks should offer support for alignment and integration. There is, however, comparatively little discussion of the *processes* that allow the EA frameworks to be used effectively in support of these goals. Here we are talking about processes in the sense of change management processes, or system development processes, rather than the actual business processes that are modelled within a system.

We propose that an additional construct, *feedback*, must be considered when assessing support for cyclic alignment. Feedback and communication exchange mechanisms are needed in an EA to provide a conduit for evolving systems and translating virtual artefacts (e.g. corporate requirements) into physical artefacts (e.g. the deployed information system). These feedback and exchange mechanisms can be uni or bi-directional and can take the form of services that allow information to be passed between several views.

In the general system sense, Sahakin (1970) states that feedback and exchange mechanisms need to be capable of 'confirming knowledge' and 'facilitating corrective action' where changes occur or faults manifest. Sahakin (1970) also asserts that feedback and exchange of information might also facilitate judgement of the necessity for change based on consequences or outcomes from adopting the change.

Overall, any adopted feedback and exchange mechanisms should be optimally connected to ensure sufficient cohesion with minimal coupling or constraint. Allowing the mechanisms to pursue and attain a suitable balance or equilibrium of connectivity will ultimately determine their success in evolving to the next stage of development and meeting environmental change.

Figure 2 illustrates the need for feedback and communication exchange mechanisms in the process of cyclic alignment involving an EA. It is necessary to communicate corporate requirements and constraints through to models constructed, but it is also necessary for information about these constructed artefacts to be fed back, and understood, by corporate management.

From the foregoing, we conclude that a good EA needs to be complete, clear, well aligned, understandable and capable of being evolved as the environment changes. Two attributes in particular are seen as necessary for cyclic alignment: (i) the inclusion of an integration structure and (ii) the

provision of feedback and exchange mechanisms. In the following sections several EA frameworks are described and evaluated with respect to these attributes.

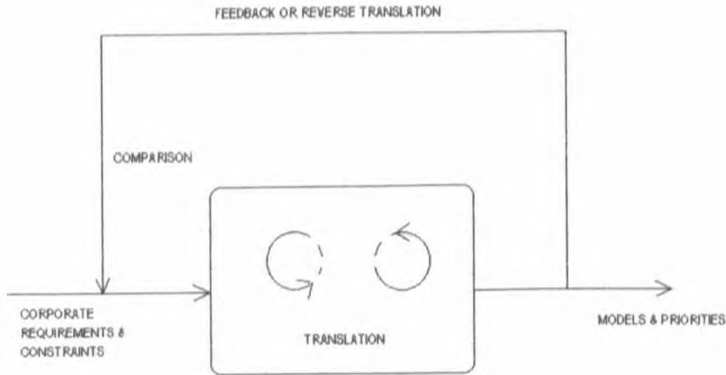


Figure 2 Feedback and exchange mechanisms needed for cyclic alignment process

Enterprise architectures

This section describes the features of a number of enterprise architecture frameworks: Zachman, CIM-OSA, ARIS and MGEA.










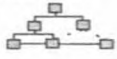




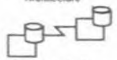
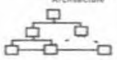














The Zachman EA

Figure 3 shows the graphic depiction of the Zachman Enterprise Architecture Framework.

The Zachman framework is built on the analogous structures that are found in the historical disciplines of public and private sector building, construction and manufacturing. These disciplines classify and organise their realised artefacts as the complex products are produced. The framework depicts the design artefacts as the interconnecting relationship between the role players in the enterprise and the product abstractions.

The Zachman framework tends to be generic in nature and may be applied to any enterprise in the private or public sector.

ENTERPRISE ARCHITECTURE - A FRAMEWORK TM

	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>	ENTITY = Class of Business Thing	Function = Class of Business Process	Node = Major Business Location	People = Major Organizations	Time = Major Business Event	Ends/Means=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 Entity Reh = 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 Entity Reh = 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. Reh = Printer/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 Reh = 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

Figure 3 The Zachman Architecture Framework

Computer Integrated Manufacture-Open Systems Architecture (CIM-OSA)

CIM-OSA is an enterprise operation improvement architecture configured as an integrated architecture (Kosanke and Vlietstra, 1989). The explicit description of the enterprise allows all internal and external processes and relationships to be mapped against the complete system description. CIM-OSA is a guiding construct for the design and delivery of the complete enterprise and all associated business functions (eg, manufacture, marketing, finance, administration, etc). Figure 4 illustrates the CIM-OSA cuboid.

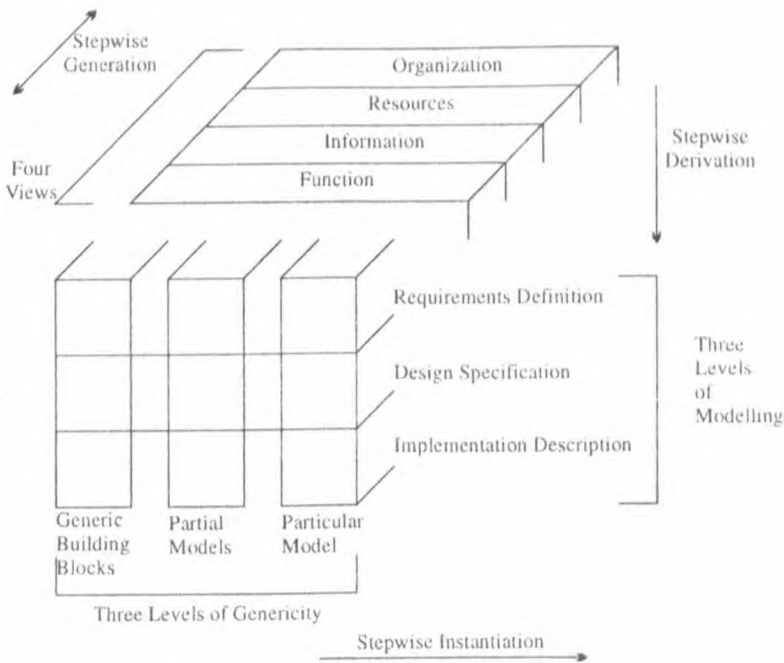


Figure 4 The CIM-OSA Framework (Kosanke and Vlietstra, 1989)

CIM-OSA has two major constructs that support enterprise integration. The Integrating Infrastructure (II) provides application integration, while the Modelling Framework (MF) supports business integration. II provides four service sets as follows:

- **Functional Services** providing management control and execution of enterprise activities. These services integrate Business Process Control, Activity Control and Resource Management.
- **Information Services** support information processing for the enterprise. These services locate, access, store and maintain information and data sets. These services fuse enterprise wide information.
- **Communication Services** control Intra and Inter system communications. These services provide integrated communications across the enterprise.
- **Front End Services** provide the interface control between communications, humans, machines and applications. These services are interface controllers and form key integration nodes.

There are three specific modelling levels that form part of the CIM-OSA framework:

- **Requirements Definition Level** that models business requirements for the complete enterprise. These requirements are depicted in terms of processes, inputs, outputs and procedural rules, describing what must be done in the business.
- **Design Specification Level** that models the design of the business processes and enterprise activities describing how the activities are performed. Parameters are specified, output size is defined and constraining factors examined.
- **Implementation Description Level** is the 'executable' level that selects the entities (ie, personnel, programs, etc) required for the process at the requirement level. The entities are selected on the basis of the design specifications and must be acquired if they are not resident. The flowdown from requirements to implementation is underpinned by computer programming (or software design).

The views that are facilitated by CIM-OSA are as follows:

- **Function view** is a depiction of the enterprise in terms of the structured business processes. Each process is constrained by its procedural rule set that is in turn defined by event triggers and results. A high level business process can be made up from a series of base level enterprise activities.

- **Information view** is the aggregate of all enterprise information. The information is decomposed into classes and enterprise information objects, with object views and editions encapsulated in domains that are determined by the design. All information is formed by information elements, the smallest addressable unit.
- **Resource view** contains all relevant information about the enterprise resources, and is formed through the hierarchical assembly of matching resources to enterprise requirements.
- **Organisational view** contains the various responsibility assignments for the enterprise and allows for view structuring in line with function, information and resource allocation. Enterprise views will be generated in sequence with program sets supporting the enterprise design process.

CIM-OSA makes the important point that people make enterprise systems work. People drive IT and manufacturing systems and not the other way round. CIM-OSA provides for Business, Application and Physical integration in two major environments:

- **Integrated Enterprise Engineering Environment** is the implementation model that is composed of the business processes and enterprise activities required to implement CIM-OSA. The implemented CIM-OSA guidelines specify the requirements, design, implementation and release of the enterprise system. The model also includes the information, resource and organisation views.
- **Integrated Enterprise Operation Environment** is the implementation model that is composed of the business processes and enterprise activities required to operate the complete enterprise, and includes the relevant information, resource and organisation views.

The creation and integration mechanism of the enterprise architecture is facilitated by the application of instantiation, derivation and generation being applied to all levels and views of the CIM-OSA cuboid. The enterprise architecture is delivered by the owner and created by the enterprise user community. The creation process is delivered using the II services that are vendor independent and provide application portability across the enterprise.

Architecture of Integrated Information Systems (ARIS)

ARIS is complementary to Zachman and is seen as a framework of views that describe the enterprise and is fully integrated by the process oriented view (Nagel, 2001). The business processes, functions, data, organisational structures and outputs are the respective ARIS views. The three active levels are the main

stages of a software engineering lifecycle – requirements definition, design specification and build-implementation. Figure 5 shows the ARIS framework.

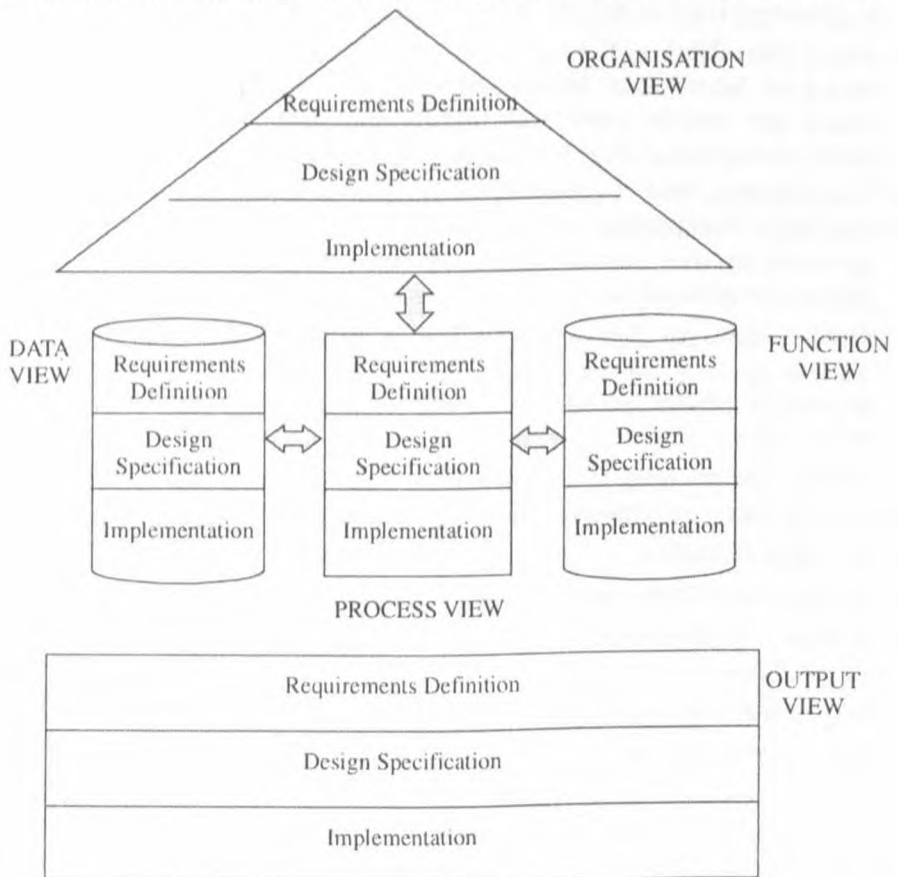


Figure 5 The ARIS Framework (Nagel, 2001)

ARIS has significant and direct concentration on business processing and accordingly, the process view prevails as the basis of integration of all elements in an Enterprise Architecture.

- **Process view.** This view shows the relationships between enterprise objectives, activities, events documents, data, organisational units, resources and knowledge sets (ie structure, logic, time). The technology model that is most popularly deployed is Event-driven Process Chains. This model is used in documenting processes in the popular SAP R/3 Enterprise System.

- **Function view.** Functions are used as descriptors for essential value creating activities for strategic business goals. Functions are the dynamic portion of the business process and are described in functional analysis outputs.
- **Data view.** Data and information are descriptors for the transformation stages of the relevant business objects. Data can form business process inputs and outputs while each transforming event can realise a data set. Entity-Relationship diagrams can be used to model the data view.
- **Organisation view.** Organisational entities (ie, team, person, role, etc) are the major components of this view, where component arrangement is governed by structure, or hierarchical rules. This view shows the resource allocations required for delivery of the tasks within each business process.
- **Output view.** An object-oriented outlook represents the ARIS output view. This perspective captures the results of the business process and realises internal or product based results. This view also provides for product and service hierarchies.

ARIS also provides Description Levels that are matched to the software engineering lifecycle. These levels are represented as follows:

- **Problem Definition** – business problem is defined
- **Requirement Definition** – user-system requirements are defined in detail
- **Design Specification** – system design document is compiled into a specification
- **Implementation Description** – the build strategy is completely described
- **Information Technology** – the technology or system is realised

The process view is the ‘prime integrator’ of the ARIS house. The process view integrates itself with the remaining four views to deliver the complex enterprise model. As noted earlier, the Event-driven process chains are the most popular modelling technique. The process view is the ‘integration concept’ that supports the view and description level integration activities.

META Group Enterprise Architecture (MGEA)

The META Group (Westbrock, 1999) have proposed an enterprise architecture strategy that commences with a set of common requirements and corporate vision, defines a set of guiding concepts, and establishes a set of domain architectures for enterprise growth and evolution. The Enterprise Information Architecture is platformed on an existing base of information and infrastructure. Figure 6 illustrates the MGEA. The most common enterprise

architecture delivered by the strategy has two specific domains termed Business and Information Technology.

The Business domain encapsulates the Operational and Business architectures and the Information architectures. The Operational and Business architectures hold the business models, business processes and organisation (human assets) artefacts. The Information architecture defines the business language in terms of defining and publishing the meaning, source, and associated business rules for the important terms used in the enterprise. The Information architecture also has the enterprise data models and relationships resident.

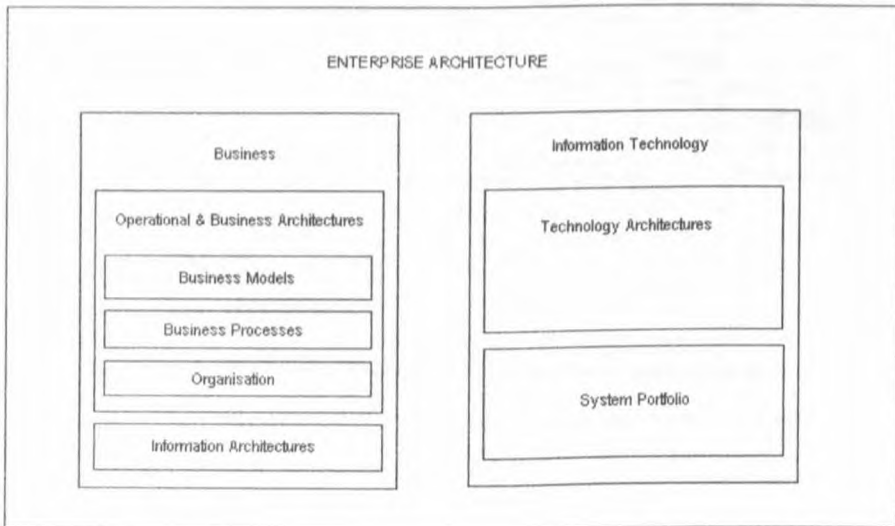


Figure 6 The MGEA Framework (Westbrock, 1999)

The Information Technology domain encapsulates the Technical architecture and the Systems Portfolio. The Technical architecture defines the principles, technologies, products and standards that support the information environment. This includes the standard operating environments, reference models, and technical standards that underpin the environment. The Systems portfolio is the collection of all enterprise information systems and includes architectural principles, application strategies, all hardware and software components, environment gap analysis, and an evolution plan and investment strategy.

The driving mechanism for enterprise architecture creation, integration and deployment is the gap analysis function in the systems portfolio. The gap analysis drives the evolution path by measuring differences between the 'as is' and 'to be' portfolio and implementing changes that address the gap. The gap analysis is also fed by the domain architectures so that changes in those domains are also reflected in the analysis and implementation activities. Figure 7 illustrates the EA creation process in MGEA.

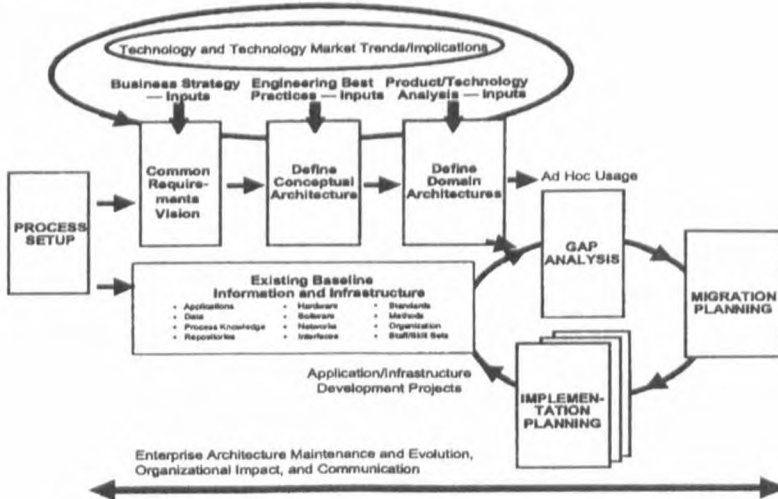


Figure 7 The EA creation process in MGEA (Westbrock, 1999)

Comparison of EAS for cyclic alignment qualities

It was argued previously that the enterprise architecture process should be dynamic and support feedback exchange and information systems development across the system life-cycle. Enterprise Integration (EI) is likely not best platformed on a static depiction or classification of separated enterprise artefacts. Enterprise models should be capable of evolution in keeping with system and environmental changes.

Each of the EAs described above was analysed in terms of the attributes contributing to cyclic alignment capability: integration mechanisms and exchange and feedback mechanisms. Table 1 shows a summary of the results.

All the aforementioned reference architectures, except for the Zachman framework, have a common theme for their creation mechanism in the form of an 'integrating structure or view'. Some of the architectures link their constituent

models with feedback and exchange channels, although some channels are unidirectional information flows. None of the selected architectures, except ARIS, have a discernible feedback loop to corporate requirements and constraints.

Table 1 Comparison of EA frameworks for cyclic alignment support

Architecture Type	Integration Mechanism	Feedback and Exchange Channels	Feedback to Corporate Requirements and Constraints
Zachman Architecture Framework	No evident integration mechanisms. Essentially a classification framework.	No evident feedback and exchange channels.	No discernible feedback channels to Corporate Requirements and Constraints.
CIM-OSA	Integration Infrastructure and Services integrate the Function, Information, Resource and Organisation views.	Integration Services channels pass information and feedback between views.	No discernible feedback channels to Corporate Requirements and Constraints.
ARIS	Process View integrates the Organisational, Data and Functional views.	Two way feedback between the Organisation, Data, Function and Process views.	Two way feedback channel between Process and the Organisational views (Requirements Definition).
MGEA	Integration Services (Gap Analysis, Migration Planning, Implementation Planning) integrate the requirements, architectures, information and infrastructure.	Unidirectional feed forward channels aggregate requirements, architectures, information and infrastructure	No discernible feedback channels to Corporate Requirements and Constraints.

Conclusions

The e-commerce community is under constant pressure to deliver new systems faster and with optimal performance. Systems engineering standards, however, indicate the use of formal development methods that are measured and structured, though they may lack the responsiveness and agility required to meet the faster paced e-Commerce environment. In this systems development environment there may be a positive role for EA with certain attributes.

The literature and empirical evidence suggest that modelling tools should be evaluated with respect to ontological completeness and clarity, support for good decomposition and simplicity and understandability. For EA frameworks we should also look for enterprise alignment and integration capabilities. We suggest here additional criteria that assess the degree to which an EA framework allows for a *cyclic alignment process*. This is a process in which corporate requirements are communicated throughout a development project and acted upon in system construction, but in addition there are feedback mechanisms to allow understanding and learning to flow back to corporate management in a cyclic process.

The ideal EA should be optimally connective, yet not overly constraining in its structure. The EA should seek to strike a balance between structural modular freedom and the alignment of models, purpose and enterprise goals. We have reviewed several proprietary EA frameworks in terms of the support they offer for integration and feedback mechanisms. All frameworks except for the Zachman architecture offer support for an integrating structure. Most link their constituent models with feedback and exchange channels, though some channels are unidirectional. None of the selected architectures, except for ARIS, has a discernible feedback loop to corporate requirements specification.

The apparent lack of research into EA artefact or model connectivity provides an avenue for further research to determine the nature and utility of feedback channels and exchanges, including their optimal structure and use.

References

1. Croteau, A-M., Solomon, S., Raymond, L., and Bergeron, F. (2001). Organisational and Technological Infrastructures Alignment. *Proceedings of the 34th Hawaii International Conference on System Sciences*.
2. Karimi, J. and Konsynski, B.R. (1991). Globalisation and Information management systems, *Journal of Management Information Systems*, 4, 7-26.
3. Kosanke, K. and Vlietstra, J. (1989) CIM-OSA – Its goals, scope, contents and achievements, *ESPRIT '89: Proceedings of the 6th Annual ESPRIT Conference*, Boston: Kluwer Academic.
4. Nagel, C. (2001). Successfully Building Enterprise Architectures – How ARIS facilitates the Zachman Framework, *Leonardo Consulting White Paper*.
5. Navathe, S.B. (1992). Evolution of data modelling for databases, *Communications of ACM*, 35, 112-23.
6. Pres-Heje, J. and Baskerville, R. (2001). eMethodology: Toward a systems development methodology for e-Business and e-Commerce applications". In S. Elliot, K. V. Andersen, P. Swatman, S. Reich (Eds.). *Developing a dynamic, integrative, multi-disciplinary research agenda in e-commerce/e-business*, International Federation of Information Processing.
7. Sahakian, W.S. (1970). *Psychology of learning: Systems, Models and Theories*. Chicago: Markham Publishing Company.
8. Vernadat, F.B. (2000). Enterprise Modelling and Integration: Myth or Reality. (adapted from *Enterprise Modelling and Integration: Principles and Applications*) (Vernadat, F.B.), Chapman and Hall, London, 1996.
9. Wand, Y. and Weber, R. (1995). On the deep structure of Information Systems, *Information Systems Journal*, 203-23.
10. Wassenaar, A., Gregor, S. and Swagerman, D. (2002, to appear). ERP Implementation management in different organizational and cultural settings, *Proceedings of European Conference on Accounting Information Systems*, 23-24 April, Athens.
11. Weber, R. (1997). *Ontological Foundations of Information Systems*. Coopers and Lybrand and the Accounting Association of Australia and New Zealand, Melbourne, Australia
12. Westbrook, T. (1999). Managing change through Enterprise Architecture, *Enterprise Architecture Strategies*, Meta Group Practice, August.
13. Weston, R.H. (1999). Reconfigurable Component-based Systems and the Role of Enterprise Engineering Concepts, MSI Research Institute Paper, Loughborough University. http://www.cit.gu.edu.au/~bernus/taskforce/mandate/weston_12febr99/weston_12febr99.html
14. Zachman, J.A. (1996). Enterprise Architecture – Issue of the Century, *Database Programming and Design Magazine*, March. (<http://www.zifa.com/zifaz01.htm>)