TOGAF/MDA Mapping

A White Paper developed by:

Pete Rivett, CTO, Adaptive, Inc.
John Spencer, Director, Architecture Forum, The Open Group
Fred Waskiewicz, OMG Director of Standards, OMG

September, 2005

TOGAF/MDA Mapping

Document No.: W054
(equivalent to OMG Document No.: omg/2005-08-01)

Published by The Open Group, September, 2005

Any comments relating to the material contained in this document may be submitted to:

The Open Group
44 Montgomery St. #960
San Francisco, CA 94104

or by email to:

mda-togaf@opengroup.org

Table of Contents

Boundaryless Information Flow™
achieved through global interoperability
in a secure, reliable, and timely manner

Executive Summary

This White Paper is one of a series of documents produced by a unique collaboration between three leading consortia in the IT industry: The Integration Consortium, OMG, and The Open Group.

Building on an earlier White Paper published jointly by OMG and The Open Group, the Paper provides a mapping between two sets of industry standards – the TOGAF Architecture Development Method (ADM) for enterprise architecture, developed by The Open Group Architecture Forum; and OMG's Model Driven Architecture (MDA) family of modeling standards. The Integration Consortium intends to use this mapping as the basis of its own Global Integration Framework (GIF), and has performed the crucial role of user of the TOGAF/MDA mapping in this collaborative effort.

This White Paper is intended for decision-makers in all enterprises seeking to develop enterprise architectures based on open industry standards. In particular, it will be of interest to CIOs, Chief Architects, and those responsible for the oversight of architecture efforts within their enterprise.

PART 1: Introduction

This White Paper is one of a series of documents produced by a unique collaboration between three leading consortia in the IT industry: The Integration Consortium, OMG, and The Open Group.

Building on an earlier White Paper published jointly by OMG and The Open Group, the Paper provides a mapping between two sets of industry standards – the TOGAF Architecture Development Method (ADM) for enterprise architecture, developed by The Open Group Architecture Forum; and OMG's Model Driven Architecture (MDA) family of modeling standards. The Integration Consortium intends to use this mapping as the basis of its own Global Integration Framework (GIF), and has performed the crucial role of user of the TOGAF/MDA mapping in this collaborative effort.

Format and Structure

The remainder of Part 1 provides an introduction to this effort, including the rationale and goals of the collaboration, and explains some of the key technical concepts involved.

Part 2 of the White Paper forms the core of the mapping. It has been developed by taking excerpts from the TOGAF documentation suite describing those phases of the TOGAF Architecture Development Method (ADM) addressing the development of architecture descriptions, and annotating them with references to relevant MDA standards. These annotations are *highlighted in red*.

Part 3 provides a summary description of all the MDA standards referenced in Part 2. Also, whereas Part 2 provides a TOGAF-centric view of the mapping, Part 3 provides an MDA-centric view.

Overview of the TOGAF ADM/MDA Synergy Activity

The Problem Statement

The vision for this activity was originally set out in a previously published White Paper.[1]

Just as architecture is critical for the effective and safe construction of buildings or bridges, so too is architecture critical to the construction of business and information systems. A proven architecture can speed up implementation, reduce total costs, and increase project success rates.

Yet organizations struggle to develop quality architectures that meet business needs.

A Solution

The Open Group Architecture Framework (TOGAF) provides the TOGAF Architecture Development Method (ADM).[2] TOGAF ADM is a comprehensive, detailed, industry standard method for developing enterprise architectures, and related information, application, and technology architectures, that address the needs of business, technology, and data systems. It calls for the development of a number of architectural models in order to effectively describe the architectures.

The Object Management Group Model Driven Architecture (MDA)[3] is the industry's leading standards-based approach to model-based business-focused architecture and systems development. It

[1] T. Blevins, J. Spencer, F. Waskiewicz, TOGAF ADM and MDA, Revision 1.1 (www.opengroup.org/cio/MDA-ADM).

[2] John Spencer, et al (2004), TOGAF Enterprise Edition, Version 8.1 (www.opengroup.org/architecture/togaf8-doc/arch).

[3] Object Management Group, MDA Guide, Version 1.0.1 (www.omg.org/docs/omg/03-06-01.pdf).

enables the modeling, planning, and development of enterprise software systems in a way that conforms to the chosen architecture. MDA can be coupled with other industry standard technologies to assist in the deployment and management of these systems. It also provides for the re-usability and easy maintainability of the components of the architecture throughout its lifecycle – supporting the cycle from business requirements through to implementation and maintenance.

Both TOGAF ADM and MDA complement the use of popular development frameworks, such as the Zachman Framework,[4] and the RM-ODP (Reference Model of Open Distributed Processing),[5] that enable real-world business concepts to be related to the concepts of information systems and their specification and implementation.

In addition to being adopted by many end-user organizations, these standards have also been endorsed by the Integration Consortium and form an integral part of its Global Integration Framework (GIF).

These vendor-neutral industry standard frameworks are complementary, and provide the potential for enormous business value if used effectively together. Accordingly, the memberships of The Open Group, the OMG, and the Integration Consortium are collaborating to describe and promulgate the synergies that they believe exist between the TOGAF Architecture Development Method (ADM) and OMG's Model Driven Architecture (MDA).

Benefits of the Solution

Specifically, this activity is envisaged to provide the following benefits:

- Provide a coordinated, disciplined approach to developing and implementing standards-based enterprise architectures and systems that are focused on meeting business needs
- Provide a clear and unambiguous process for:
  - Creating enterprise architectures and their related detailed architectures and system designs
  - Creating higher-quality and lower-cost deliverables
  - Re-use of architecture assets
  - Standards-based traceability from the operations level back to architecture
- Provide a single source point for technology and processes related to developing enterprise architectures
- Enable architecture practitioners to select the elements of TOGAF and MDA that will provide a disciplined approach to the development of architectures and systems that are tailored to the specific business needs of the organization
- Provide clearly defined, industry-wide roles and accountability for the effective creation of architectures and systems
- Promote a higher degree of transparency that will improve the communications and understanding between architecture, development, and operations personnel
- Enhance the portability, re-usability, and interoperability of enterprise architectures and system architectures to facilitate the efficient and effective integration of enterprise applications
- Provide opportunities for tool vendors to create products that leverage a proven chain of methods and standards without having to cover the whole span from business rationale to running code

[4] J.F. Sowa, J.A. Zachman, Extending and Formalizing the Framework for Information Systems Architecture, IBM System Journal, Volume 31, No 3, 1992.

[5] IITU-T Recommendations X.901-904 | ISO/IEC 10746, Reference Model of Open Distributed Processing (RM-ODP).

- Promote greater collaboration among consultancies and system integrators and their subcontractors in managing large, complex projects through leveraging well articulated methods and guidelines
- Demonstrate to architecture practitioners that TOGAF ADM and MDA can be used together with a framework of choice (such as the well-known Zachman Framework), to bring greater discipline and re-usability to enterprise architecture

It is also envisaged that the activity to articulate the synergies between TOGAF and MDA will produce improvements to both that will enable organizations to develop even better quality architectures.

Clearly, there are some very good reasons to ensure that TOGAF and MDA can work together when desired. The value proposition for the members of The Open Group, OMG, and the Integration Consortium will be that the whole will be greater than the sum of the parts. For enterprise architects, the value proposition will be a more seamless architecture model that spans several layers of architecture planning from the highest level business modeling to the most detailed systems specifications.

Over the next 9 to 18 months, the combined contributing organizations and their respective members intend to demonstrate, concretely, these posited synergies.

Planned Deliverables from the Collaboration

The purpose of this activity is to develop a formal, model-driven, industry-independent, standards-based enterprise architecture development methodology, a framework for enterprise architectures, and an information sharing framework. These technologies will be grounded in a formal modeling approach and framework that defines and creates architectural artifacts (e.g., models) enabling a disciplined approach to enterprise architecture and engineering.

Rather than simply being a theoretical exercise, the conclusions drawn from this activity will be applied to actual test cases drawn from industry. To that end, the Integration Consortium and the Torpedo Enterprise Advanced Modeling and Simulation (TEAMS) Initiative are participating as commercial, government, and defense contributors.

Specific planned deliverables include:

- A set of standards:
  - Enabling the use of a repeatable architecture methodology across all projects/programs/portfolios, ensuring integration and interoperability
  - Enabling technology development, evaluation, and selection methods to be mapped to business goals and strategies
  - Promoting re-use
  - Supporting increased system development flexibility/agility/adaptability, etc. in a cohesive and disciplined way
- Guidelines and descriptive business best practices providing methods and procedures for improved product/service quality from concept to implementation
- Reports and White Papers
- Various services that may include (among others) certification of implemented standards

An Invitation

The Open Group, OMG, and the Integration Consortium are inviting senior technologists within their membership with an interest and experience in developing enterprise architectures, largescale systems, and/or systems-of-systems to participate in this activity.

Participants (ideally) will have an understanding of TOGAF ADM and/or MDA. Lacking that, they should be conversant in architecture and systems development methodologies and a model-driven approach to software specification, design, implementation, and maintenance. Participation can take many forms; e.g., developing proofs of concept, modeling, creating test cases, or writing White Papers.

Interested parties should email expressions of interest to: mda-adm@opengroup.org.

TOGAF/MDA Mapping of Concepts

MDA Generic Concepts

The official definition of MDA is contained in the MDA Guide.[6] The key messages regarding MDA can be summarized as follows:

- Business requirements and information technology design are captured in models.
- MDA relies on the modeling technique of abstraction to separate business concerns from technology concerns (what the business system needs to do, *versus* how its underlying computing platform does it).
- The purpose of a Computation-Independent Model (CIM) is to capture business requirements in the language that is "natural" to the end user of the IT solution.
- A Platform-Independent Model (PIM) is intended to specify an IT solution that is not tied to the choice of a specific computing platform. This promotes designing a business solution prior to selecting how it will be deployed.
- The Platform-Specific Model (PSM) adds to the PIM the details of a specific computing platform on which the business solution will be deployed.
- To achieve a complete computing solution to the business problem stated as requirements, MDA provides for automated code generation from the design models.
- Transformations (mappings) are performed on these models to progress from a higher level of abstraction to a lower level of abstraction.
- Each of the above classes of model is viewed in the context of its platform (e.g., normalized data, relational database, Oracle database), hence one person's PSM may be another person's PIM.
- All of this activity is based on internationally accepted standards.

[6] Object Management Group MDA Guide, Version 1.0.1 (www.omg.org/docs/omg/03-06-01.pdf).

Generic Mapping of MDA and the TOGAF ADM Cycle

Figure 1 shows the basic structure of the TOGAF ADM cycle, and how the generic concepts of MDA map on to it.

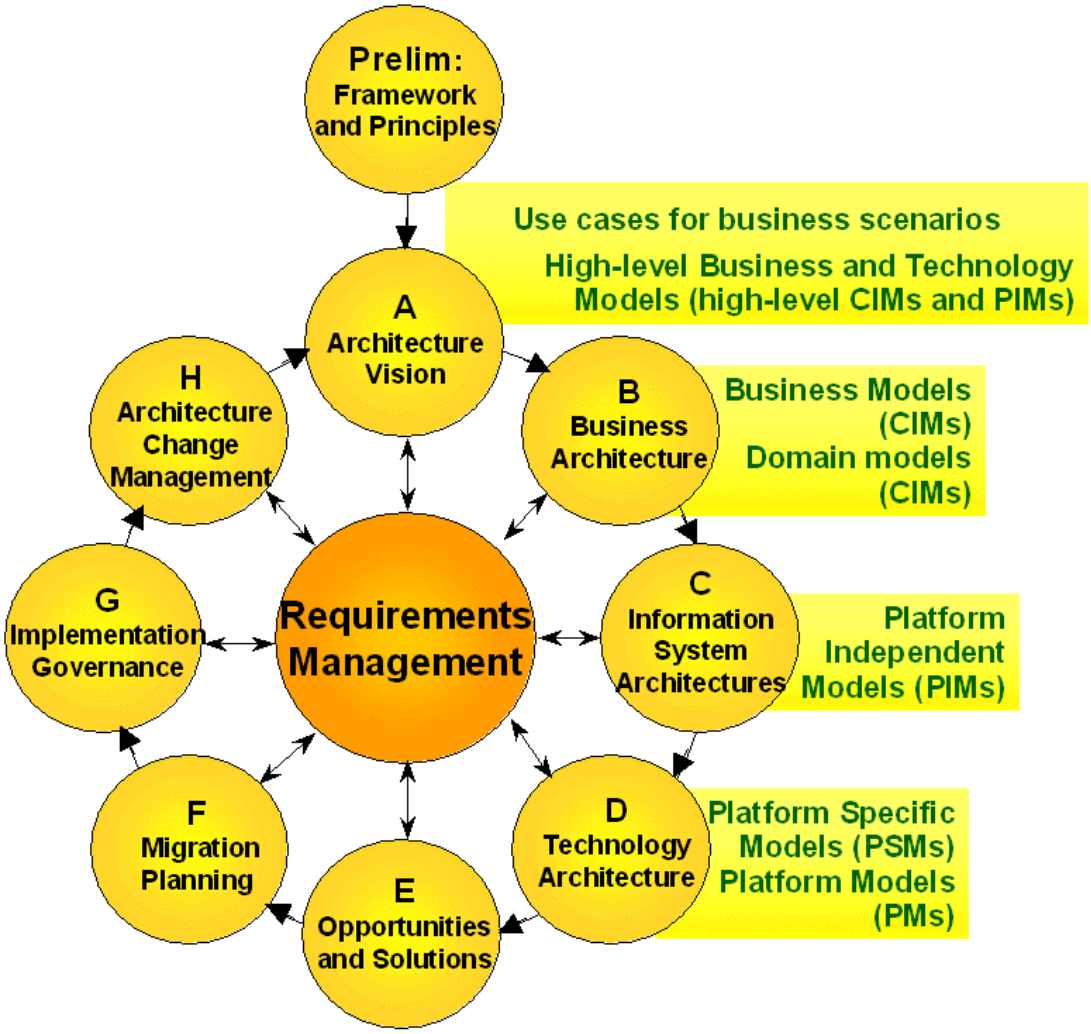


Figure 1: ADM Lifecycle and Relevant Generic MDA Concepts

Key Synergy Points

Figure 2 below shows in more detail the key synergy points between the TOGAF ADM and MDA, in terms of how the generic concepts of MDA map to specific Phases of the ADM, in terms of three criteria: Applicable, Useful, and Not Applicable.

	CIM	PIM	PSM	PM	Transformations
ADM Phase A	A	A	N/A	N/A	A
ADM Phase B	A	A	N/A	N/A	A
ADM Phase C	N/A	A	N/A	N/A	A
ADM Phase D	N/A	N/A	A	A	A
ADM Phase E	N/A	N/A	U	N/A	A (if PSM created)
ADM Phase F	N/A	N/A	N/A	N/A	N/A
ADM Phase G	N/A	N/A	U	U	A (if PSM & PM created)
ADM Phase H	N/A	N/A	N/A	N/A	N/A

Figure 2: *TOGAF ADM/MDA Synergy Points*

A: Applicable. TOGAF specifies that a model of this type be created in this phase.

N/A: Not Applicable. TOGAF does not specify that a model of this type be created in this phase.

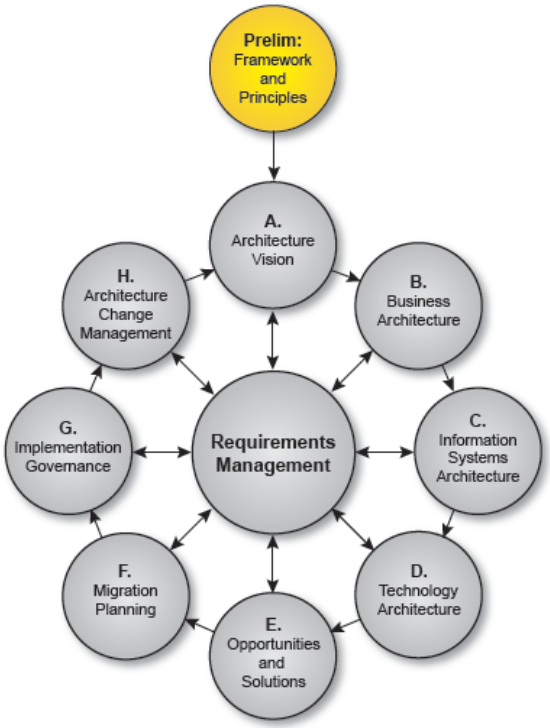
U: Useful. TOGAF does not specify that a model of this type be created in this phase, but creating such a model might provide useful information.

The different phases of the ADM cycle, and the relevance of MDA concepts in each phase, are described in detail in the following sections, in which the descriptions of the various ADM phases are reproduced, with MDA-specific material indicated and *highlighted in red*.

PART 2: Mapping the TOGAF ADM to MDA Standards

Preliminary Phase: Framework and Principles

Objectives



- To ensure that everyone who will be involved in, or benefit from the architectural process is committed its success
- To define the architecture principles that will inform the constraints on any architecture work

  Things important to the business (such as principles) could be represented by the Business Motivation Metamodel (BMM).
- To define the "architecture footprint" for the organization – the people responsible for performing architecture work, where they are located, and their responsibilities

  These organization aspects can be represented using the Organization Structure Metamodel (OSM).
- To define the scope and assumptions (particularly in a federated architecture environment)
- To define the framework and detailed methodologies that are going to be used to develop enterprise architectures in the organization concerned (typically, an adaptation of the generic ADM)

  The Software Process Engineering Metamodel (SPEM) can be used to model the TOGAF process itself.
- To set up and monitor a process (normally including a pilot project) to confirm the fitness-for-purpose of the defined framework
- If necessary, to define a set of criteria for evaluating architecture tools (an example set of criteria is given in Part IV: Resource Base), repositories, and repository management processes to be used to capture, publish, and maintain architecture artifacts

  Ability to support MDA standards could be part of the criteria.

Approach

This Preliminary Phase is about defining "how we do architecture" in the enterprise concerned. There are two main aspects: defining the framework to be used; and defining the architecture principles that will inform any architecture work.

The enterprise's approach to re-use of architecture assets is a key part of both the framework definition

and architecture principles. (Typically the principles will state the policy on re-use; and the framework will explain how re-use is effected.)

The Reusable Asset Specification (RAS) could be used to model re-use aspects of all the architectural assets held, and their inter-relationships, and provide access to the specific models and artifacts.

In federated architectures (see Part II: ADM, Introduction to the ADM, Enterprise Scope/Focus), requirements from a higher-level architecture are often manifested as "principles" in lower-level architectures.

Principles

The Preliminary Phase defines the architecture principles that will form part of the constraints on any architecture work undertaken in the enterprise. The issues involved in this are explained in Part IV: Resource Base, Architecture Principles.

Architecture work is informed by business principles as well as architecture principles. The architecture principles themselves are also normally based in part on business principles. Defining business principles normally lies outside the scope of the architecture function. However, depending on how such principles are defined and promulgated within the enterprise concerned, it may be possible for the set of architecture principles to also restate, or cross-refer to, a set of business principles, business goals, and strategic business drivers defined elsewhere within the enterprise. (Within an architecture project, the architect will normally need to ensure that the definitions of these business principles, goals, and strategic drivers are current, and to clarify any areas of ambiguity.)

The issue of architecture governance is closely linked to that of architecture principles. The body responsible for governance will also normally be responsible for approving the architecture principles, and for resolving architecture issues. This will normally be one of the principles cited. The issues involved in governance are explained in Part IV: Resource Base, Architecture Governance.

Principles can be represented as Policy elements in the Business Motivation Metamodel (BMM).

Framework

The TOGAF Architecture Development Method (ADM) is a generic method, intended to be used by enterprises in a wide variety of industry types and geographies. It is also designed for use with a wide variety of other enterprise architecture frameworks, if required (although it can be used perfectly well in its own right, without adaptation).

The Preliminary Phase therefore involves doing any necessary work to adapt the ADM to define an organization-specific framework, using either the TOGAF deliverables or the deliverables of another framework. The issues involved in this are discussed in Part II: ADM, Introduction to the ADM, Adapting the ADM.

MDA, and this mapping document, would be an input to such an adaptation exercise.

The Software Process Engineering Metamodel (SPEM) can be used to model the TOGAF process itself (and the adapted, organization-specific version of it).

Inputs

The inputs to the Preliminary Phase are:

- TOGAF Architecture Development Method (ADM)
- Other architecture framework(s), if required
- Business strategy, business principles, business goals, and business drivers (when pre-existing)

  These can all be represented by the Business Motivation Metamodel (BMM).
- IT governance strategy (when pre-existing)
- Architecture principles (when pre-existing)

  Principles can be represented as Policy elements in the Business Motivation Metamodel (BMM).
- Principles that are being subscribed to, arising from other, federated architectures.

  Principles can be represented as Policy elements in the Business Motivation Metamodel (BMM).

Steps

The TOGAF ADM is a generic method, intended to be used by a wide variety of different enterprises, and in conjunction with a wide variety of other architecture frameworks, if required. It is not practical to define specific steps for adapting the ADM to such a wide variety of potential contexts. The issues involved are discussed in detail in Part II: ADM, Introduction to the ADM, Adapting the ADM.

Outputs

The outputs of the Preliminary Phase are:

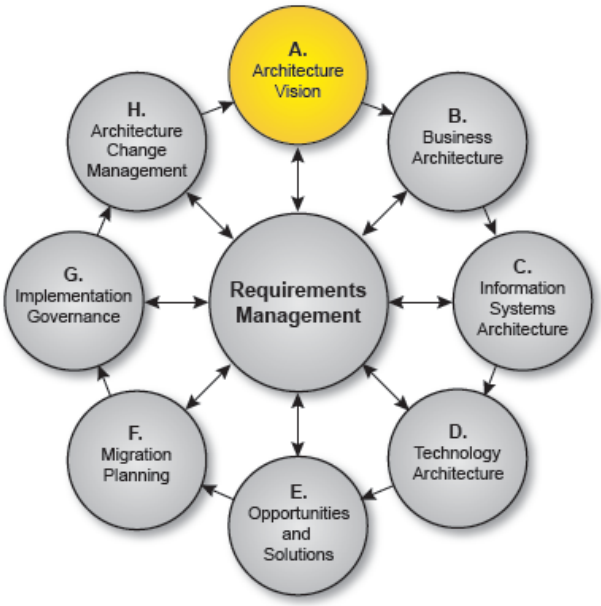
- Framework definition

  Potentially modeled using the Software Process Engineering Metamodel (SPEM).
- Architecture principles

  Principles can be represented as Policy elements in the Business Motivation Metamodel (BMM).
- Restatement of, or reference to, business principles, business goals, and business drivers

  Principles can be represented as Policy elements in the Business Motivation Metamodel (BMM).

Phase A: Architecture Vision



Objectives

- To ensure that this evolution of the architecture development cycle has proper recognition and endorsement from the corporate management of the enterprise, and the support and commitment of the necessary line management
- To validate the business principles, business goals, and strategic business drivers of the organization

  All these aspects can be represented by the Business Motivation Metamodel (BMM). In addition, business vocabulary can be captured using Business Semantics of Business Rules (BSBR) and (information) concepts can be represented using the Ontology Definition Metamodel (ODM).

  Modeling these would help the validation process.
- To define the scope of, and to identify and prioritize the components of, the current architecture effort

  Modeling the ADM process (the generic process and/or the adapted, organization-specific one), by means of the Software Process Engineering Metamodel (SPEM), would help the scoping definition process.
- To define the relevant stakeholders, and their concerns and objectives

  The Business Motivation Metamodel (BMM) can capture these; where necessary this can be linked to the Organization Structure Metamodel (OSM) for capturing structure and relationships.
- To define the key business requirements to be addressed in this architecture effort, and the constraints that must be dealt with

  Could be modeled by SysML, which (in the form of the current submission) has a facility for defining business requirements using UML.
- To articulate an architectural vision that demonstrates a response to those requirements and constraints

  IT Portfolio Management Facility (ITPMF) is an integrating metamodel that pulls together technical and business metamodels. It is a candidate for use in developing high-level models that span both the business and IT domains (as with an "architectural vision").
- To secure formal approval to proceed

  Software Process Engineering Metamodel (SPEM) can model the approval process.
- To understand the impact on, and of, other enterprise architecture development cycles ongoing

in parallel

Holding these other enterprise architecture development cycles as SPEM models would greatly help this impact assessment.

Approach

Phase A starts with receipt of a Request for Architecture Work from the sponsoring organization to the architecture organization.

The issues involved in ensuring proper recognition and endorsement from corporate management, and the support and commitment of line management, are discussed in Part IV: Resource Base, Architecture Governance.

Phase A also defines what is in and what is outside the scope of the architecture effort and the constraints that must be dealt with. Scoping decisions need to be made on the basis of a practical assessment of resource and competence availability, and the value that can realistically be expected to accrue to the enterprise from the chosen scope of architecture work. The issues involved in this are discussed in Part II: ADM, Introduction to the ADM, Scoping the Architecture.

The constraints will normally be informed by the business principles and architecture principles, developed as part of the Preliminary Phase.

Normally, the business principles, business goals, and strategic drivers of the organization are already defined elsewhere in the enterprise. If so, the activity in Phase A is involved with ensuring that existing definitions are current, and clarifying any areas of ambiguity. Otherwise, it involves defining these essential items from scratch.

- *Business principles, business goals, and strategic drivers could be modeled using the Business Motivation Metamodel (BMM).*
- *Modeling these items would help the clarification process, and in particular would highlight any ambiguities or conflicts that involved trade-offs and/or prioritization by business management.*

Similarly, the architecture principles that inform the constraints on architecture work will normally have been defined in the Preliminary Phase. The activity in Phase A is concerned with ensuring that the existing principles definitions are current, and clarifying any areas of ambiguity. Otherwise, it entails defining the architecture principles from scratch, as explained in Part IV: Resource Base, Architecture Principles.

Modeling the architecture principles would similarly help clarification, and highlight any ambiguities or conflicts that involved trade-offs and/or prioritization. They can be modeled as Policy elements in the Business Motivation Metamodel (BMM).

Creating the Architecture Vision

The Architecture Vision is essentially the architect's “elevator pitch” – the key opportunity to sell the benefits of the proposed development to the decision-makers within the enterprise. The goal is to articulate an Architecture Vision that enables the business goals, responds to the strategic drivers, conforms with the principles, and addresses the stakeholder concerns and objectives.

Clarifying and agreeing the purpose of the architecture effort is one of the key parts of this activity, and the purpose needs to be clearly reflected in the vision that is created. Architecture projects are

often undertaken with a specific purpose in mind – a specific set of business drivers that represent the return on investment for the stakeholders in the architecture development. Clarifying that purpose, and demonstrating how it will be achieved by the proposed architecture development, is the whole point of the Architecture Vision.

Normally, key elements of the Architecture Vision – such as the enterprise mission, vision, strategy, and goals – have been documented as part of some wider business strategy or enterprise planning activity that has its own lifecycle within the enterprise. In such cases, the activity in Phase A is concerned with verifying and understanding the documented business strategy and goals, and possibly bridging between the enterprise strategy and goals on the one hand, and the strategy and goals on the part of the enterprise that lie within the scope of the current architecture project.

Holding these items as models will help the process of bridging between hierarchies of strategies and goals within the enterprise. Use could be made of model versioning technologies (there is a MOF Versioning specification under submission) to manage as-is and to-be versions of model(s) and plan/track evolution.

In other cases, little or no Business Architecture work may have been done to date. In such cases, there will be a need for the architecture team to research, verify, and gain buy-in to the key business objectives and processes that the architecture is to support. This may be done as a free-standing exercise, either preceding architecture development, or as part of the Preliminary Phase.

The Architecture Vision includes a first-cut, high-level description of the baseline ("as-is") and target ("to-be") environments, from both a business and a technical perspective. These outline descriptions are then built on in subsequent phases.

Holding these items as models from the start will help the process of evolution and ensure compatibility with earlier/higher-level models as successively more detail is defined.

Business scenarios are an appropriate and useful technique to discover and document business requirements, and to articulate an Architecture Vision that responds to those requirements. Business scenarios are described in Part IV: Resource Base, Business Scenarios.

- *UML Use-Cases could be used to create business scenarios, or used instead of business scenarios.*
- *The IT Portfolio Management Facility (ITPMF) metamodel pulls together technical and business metamodels, and is a candidate for use in developing high-level models that span both the business and IT domains (as with an "Architecture Vision").*

Once an Architecture Vision is defined and documented in the Statement of Architecture Work, it is critical to use it to build a consensus, as described in Part IV: Resource Base, Architecture Governance. Without this consensus it is very unlikely that the final architecture will be accepted by the organization as a whole. The consensus is represented by the sponsoring organization signing the Statement of Architecture Work.

Inputs

The inputs to Phase A are:

- Request for Architecture Work
- Business strategy, business principles, business goals, and business drivers (when pre-existing)

Could be modeled using the Business Motivation Metamodel (BMM).

- Architecture principles (when pre-existing)

  Could be modeled using the Business Motivation Metamodel (BMM).

- Enterprise Continuum – existing architectural documentation (framework description, architectural descriptions, existing baseline descriptions, etc.)

  Could take the form of existing MDA models.

  The Reusable Asset Specification (RAS) could be used to model all the architectural assets held, and their inter-relationships, with links to the specific models.

Steps

Key steps in Phase A include:

1 **Project Establishment**: Conduct the necessary (enterprise-specific) procedures to secure enterprise-wide recognition of the project, the endorsement of corporate management, and the support and commitment of the necessary line management. Include reference to the IT governance framework for the enterprise, explaining how this project relates to that framework.

2. **Business Principles, Business Goals, and Business Drivers**: Identify the business principles, business goals, and strategic drivers of the organization.

   If these have already been defined elsewhere within the enterprise, ensure that the existing definitions are current, and clarify any areas of ambiguity. Otherwise, go back to the originators of the Statement of Architecture Work and work with them to define these essential items from scratch and secure their endorsement by corporate management.

   Business principles, business goals, and strategic drivers could be modeled using the Business Motivation Metamodel (BMM).

3. **Architecture Principles**: Review the principles under which the current architecture is to be developed. Architecture principles are normally based on the business principles developed as part of the Preliminary Phase. They are explained, and an example set given, in Part IV: Resource Base, Architecture Principles. Ensure that the existing definitions are current, and clarify any areas of ambiguity. Otherwise, go back to the body responsible for architecture governance and work with them to define these essential items from scratch and secure their endorsement by corporate management.

   Architecture principles could be represented by Policy elements in the Business Motivation Metamodel (BMM).

4. **Scope**: Define what is inside and what is outside the scope of the current architecture effort. The issues involved in this are discussed in Part II: ADM, Introduction to the ADM, Scoping the Architecture. In particular, define:

   - The breadth of coverage of the enterprise
   - The level of detail to be defined
   - The specific architecture domains to be covered (Business, Data, Applications, Technology)
   - The extent of the time horizon aimed at, plus the number and extent of any intermediate time horizons
   - The architectural assets to be leveraged, or considered for use, from the organization's Enterprise Continuum

- Assets created in previous iterations of the ADM cycle within the enterprise
- Assets available elsewhere in the industry (other frameworks, systems models, vertical industry models, etc.)

The Reusable Asset Specification (RAS) could be used to model all the architectural assets held, and their inter-relationships.

5. **Constraints**: Define the constraints that must be dealt with, including enterprise-wide constraints and project-specific constraints (time, schedule, resources, etc.). The enterprise-wide constraints may be informed by the business and architecture principles developed in the Preliminary Phase or clarified as part of Phase A.

   Potentially represented by Policy and Rule elements in the Business Motivation Metamodel (BMM).

6. **Stakeholders and Concerns, Business Requirements, and Architecture Vision**: Identify the key stakeholders and their concerns/objectives; define the key business requirements to be addressed in this architecture effort; and articulate an Architecture Vision that will address the requirements, within the defined scope and constraints, and conforming with the business and architecture principles.
   - Business scenarios are an appropriate and useful technique to discover and document business requirements, and to articulate an Architecture Vision that responds to those requirements. Business scenarios may also be used at more detailed levels of the architecture work (e.g., in Phase B), and are described in Part IV: Resource Base, Business Scenarios.

     UML Use-Cases could be used to create business scenarios, or as an alternative to their use, SysML has a facility for defining business and technical requirements.
   - If the business scenario technique is used, this step will generate the first, very high-level definitions of the as-is (baseline) and to-be (target) environments, from both a business and technical perspective:
     - Baseline Business Architecture, Version 1
     - Baseline Technology Architecture, Version 1
     - Target Business Architecture, Version 1
     - Target Technology Architecture, Version 1

     The IT Portfolio Management Facility (ITPMF) metamodel could be used to pull together technical and business metamodels.

7. **Statement of Architecture Work and Approval**: Based on the purpose, focus, scope, and constraints, determine which architecture domains should be developed, to what level of detail, and which architecture views should be built. Estimate the resources needed, develop a roadmap and schedule for the proposed development, and document all these in the Statement of Architecture Work. Secure formal approval of the Statement of Architecture Work under the appropriate governance procedures.
   - A critical evaluation of the architectural starting point – the "as-is" environment described in the Architecture Vision – may be a key element of such a statement of work, especially for an incremental approach, where neither the architecture development nor the system implementation starts from scratch.

Business Scenarios

The ADM has its own method (a "method-within-a-method") for identifying and articulating the business requirements implied in new business functionality to address key business drivers, and the

implied Technology Architecture requirements. This process is known as business scenarios, and is described in detail in Part IV: Resource Base, Business Scenarios. The technique may be used iteratively, at different levels of detail in the hierarchical decomposition of the Business Architecture. The generic business scenario process is as follows:

1. **Problem**: Identify, document, and rank the problem that is driving the project.
2. **Business and Technical Environments**: Document, as high-level architecture models, the business and technical environment where the problem situation is occurring.
3. **Objectives and Measures of Success**: Identify and document desired objectives, the results of handling the problems successfully.
4. **Human Actors**: Identify human actors and their place in the business model, the human participants, and their roles.
5. **Computer Actors**: Identify computer actors and their place in the technology model, the computing elements, and their roles.
6. **Roles and Responsibilities**: Identify and document roles, responsibilities, and measures of success per actor, the required scripts per actor, and the desired results of handling the situation properly.
7. **Refine**: Check for fitness-for-purpose of inspiring subsequent architecture work, and refine only if necessary.

Outputs

The outputs of Phase A are:

- Approved Statement of Architecture Work/Project Definition, including in particular:
  - Scope and constraints
  - Plan for the architectural work
- Refined statements of business principles, business goals, and strategic drivers
  - *Potentially represented by very general UML class diagrams.*
  - *Could be modeled using the Business Motivation Metamodel (BMM).*
- Architecture principles (if not pre-existing)
  - *Potentially represented by Business Motivation Metamodel (BMM) (policy elements).*
- Architecture Vision
  - *Potentially represented using the IT Portfolio Management Facility (ITPMF).*
- Business scenario, including:
  - Baseline Business Architecture, Version 1

    MDA standards: Business Motivation Metamodel (BMM); Business Process Definition Metamodel (BPDM); Organization Structure Metamodel (OSM); Business Semantics of Business Rules (BSBR)
  - Baseline Technology Architecture, Version 1

    MDA standards: IT Portfolio Management Facility (ITPMF); UML; Enterprise Distributed Object Computing (EDOC); Enterprise Application Integration (EAI)
  - Target Business Architecture, Version 1

    MDA standards: Business Motivation Metamodel (BMM); Business Process Definition

Metamodel (BPDM); Organization Structure Metamodel (OSM); Business Semantics of Business Rules (BSBR)

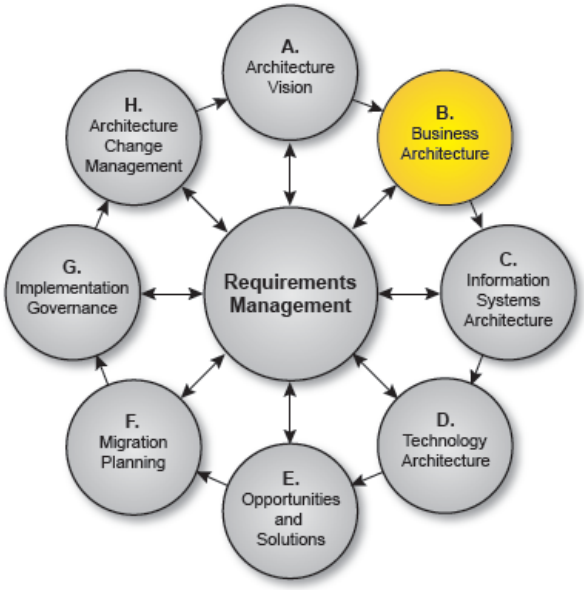
- Target Technology Architecture, Version 1

  MDA standards: IT Portfolio Management Facility (ITPMF); UML; Enterprise Distributed Object Computing (EDOC); Enterprise Application Integration (EAI)

Note: Multiple business scenarios may be used to generate a single Architecture Vision. In TOGAF terms, the Architecture Vision may also be referred to as a "conceptual-level architecture".

Phase B: Business Architecture

Objectives



- To describe the current Baseline Business Architecture
- To develop a Target Business Architecture, describing the product and/or service strategy, and the organizational, functional, process, information, and geographic aspects of the business environment, based on the business principles, business goals, and strategic drivers

  MDA standards: Organization Structure Metamodel (OSM); Business Motivation Metamodel (BMM); Business Process Definition Metamodel (BPDM); Enterprise Organization Metamodel; IT Portfolio Management Facility (ITPMF)
- To analyze the gaps between the Baseline and Target Business Architectures
- To select the relevant architectural viewpoints that will enable the architect to demonstrate how the stakeholder concerns are addressed in the Business Architecture
- To select the relevant tools and techniques to be used in association with the selected viewpoints

Approach

A knowledge of the Business Architecture is a prerequisite for architecture work in any other domain (Data, Applications, Technology), and is therefore the first architecture activity that needs to be undertaken, if not catered for already in other organizational processes (enterprise planning, strategic business planning, business process re-engineering, etc.).

In practical terms, the Business Architecture is also often necessary as a means of demonstrating the business value of subsequent Technology Architecture work to key stakeholders, and the return on investment to those stakeholders from supporting and participating in the subsequent work.

The extent of the work in Phase B will depend to a large extent on the enterprise environment. In some cases, key elements of the Business Architecture may be done in other activities; for example, the enterprise mission, vision, strategy, and goals may be documented as part of some wider business strategy or enterprise planning activity that has its own lifecycle within the enterprise.

In such cases, there may be a need to verify and update the currently documented business strategy and plans, and/or to bridge between high-level business drivers, business strategy, and goals on the one hand, and the specific business requirements that are relevant to this architecture development effort. (The business strategy typically defines what to achieve – the goals and drivers, and the metrics for success – but not how to get there. That is the role of the Business Architecture.)

In other cases, little or no Business Architecture work may have been done to date. In such cases,

there will be a need for the architecture team to research, verify, and gain buy-in to the key business objectives and processes that the architecture is to support. This may be done as a free-standing exercise, either preceding architecture development, or as part of Phase A.

In both of these cases, the business scenario technique of the TOGAF ADM, or any other method that illuminates the key business requirements and indicates the implied technical requirements for IT architecture, may be used.

Another such method, which may be used in support of or instead of business scenarios, is UML Use-Cases.

A key objective is to re-use existing material as much as possible. In architecturally more mature environments, there will be existing architecture definitions, which (hopefully) will have been maintained since the last architecture development cycle. Where existing architectural descriptions exist, these can be used as a starting point, and verified and updated if necessary (see Part III: Enterprise Continuum, The Architecture Continuum).

Gather and analyze only that information that allows informed decisions to be made relevant to the scope of this architecture effort. If this effort is focused on the definition of (possibly new) business processes, then Phase B will necessarily involve a lot of detailed work. If the focus is more on the Target Architectures in other domains (data/information, application systems, infrastructure) to support an essentially existing Business Architecture, then it is important to build a complete picture in Phase B without going into unnecessary detail.

Developing the Baseline Description

In architecturally more mature environments, there will be existing architecture definitions, which (hopefully) will have been maintained since the last architecture development cycle. Where these architectural descriptions exist, they can be used as a starting point, and verified and updated if necessary. Any such existing descriptions will already have been used in Phase A in developing an Architecture Vision, and this work should provide a sound basis for the Baseline Description, and may even be sufficient in itself.

Where no such descriptions exist, information will have to be gathered in whatever format comes to hand.

The normal approach to Target Architecture development is top-down. In the Baseline Description, however, the analysis of the current state often has to be done bottom-up, particularly where little or no existing architecture assets exist. In such a case, the architect simply has to document the working assumptions about high-level architectures, and the process is one of gathering evidence to turn the working assumptions into fact, until the law of diminishing returns sets in.

Business processes that are not to be carried forward have no intrinsic value. However, when developing baseline descriptions in other architecture domains, architectural components (principles, models, standards, and current inventory) that are not to be carried forward may still have an intrinsic value, and an inventory may be needed in order to understand the residual value (if any) of those components.

Whatever the approach, the goal should be to re-use existing material as much as possible, and to gather and analyze only that information that allows informed decisions to be made regarding the Target Business Architecture. It is important to build a complete picture without going into

unnecessary detail.

Business Modeling

A variety of modeling tools and techniques may be employed, if deemed appropriate (bearing in mind the above caution not to go into unnecessary detail). For example:

- **Activity Models** (also called **Business Process Models**[7]) describe the functions associated with the enterprise's business activities, the data and/or information exchanged between activities (internal exchanges), and the data and/or information exchanged with other activities that are outside the scope of the model (external exchanges). Activity models are hierarchical in nature. They capture the activities performed in a business process, and the ICOMs (inputs, controls, outputs, and mechanisms/resources used) of those activities. Activity models can be annotated with explicit statements of business rules, which represent relationships among the ICOMs.[8] For example, a business rule can specify who can do what under specified conditions, the combination of inputs and controls needed, and the resulting outputs. One technique for creating activity models is the IDEF (Integrated Computer Aided Manufacturing (ICAM) DEFinition) modeling technique.

  The Business Process Management Initiative (BPMI.org) is an organization[9] that is defining standards for business process modeling, including a language with which to specify business processes, their tasks/steps, and the documents produced.

- **Use-Case Models** can describe either business processes or systems functions, depending on the focus of the modeling effort. A use-case model describes the business processes of an enterprise in terms of use-cases and actors corresponding to business processes and organizational participants (people, organizations, etc.). The use-case model is described in use-case diagrams and use-case specifications.

  Note: UML has nothing about linking use-cases to business processes.

- **Class Models** are similar to logical data models. A class model describes static information and relationships between information. A class model also describes informational behaviors. Like many of the other models, it also can be used to model various levels of granularity. Depending on the intent of the model, a class model can represent business domain entities or systems implementation classes. A business domain model represents key business information (domain classes), their characteristics (attributes), their behaviors (methods or operations), and relationships (often referred to as multiplicity, describing how many classes typically participate in the relationship), and cardinality (describes required or optional participation in the relationship).[10] Specifications further elaborate and detail information that cannot be represented in the class diagram.

  The Ontology Definition Metamodel (ODM) provides a richer capability for conceptual modeling.

[7] Note: OMG does not call Activity Models Business Process Models – for the latter there is BPDM and the BPMN notation to be adopted by RFC.
[8] Note: This is IDEF0, not a UML Activity Model.
[9] Note: BPMI is now merged into OMG.
[10] Note: UML uses the term "multiplicity" for both "how many" and "optionality".

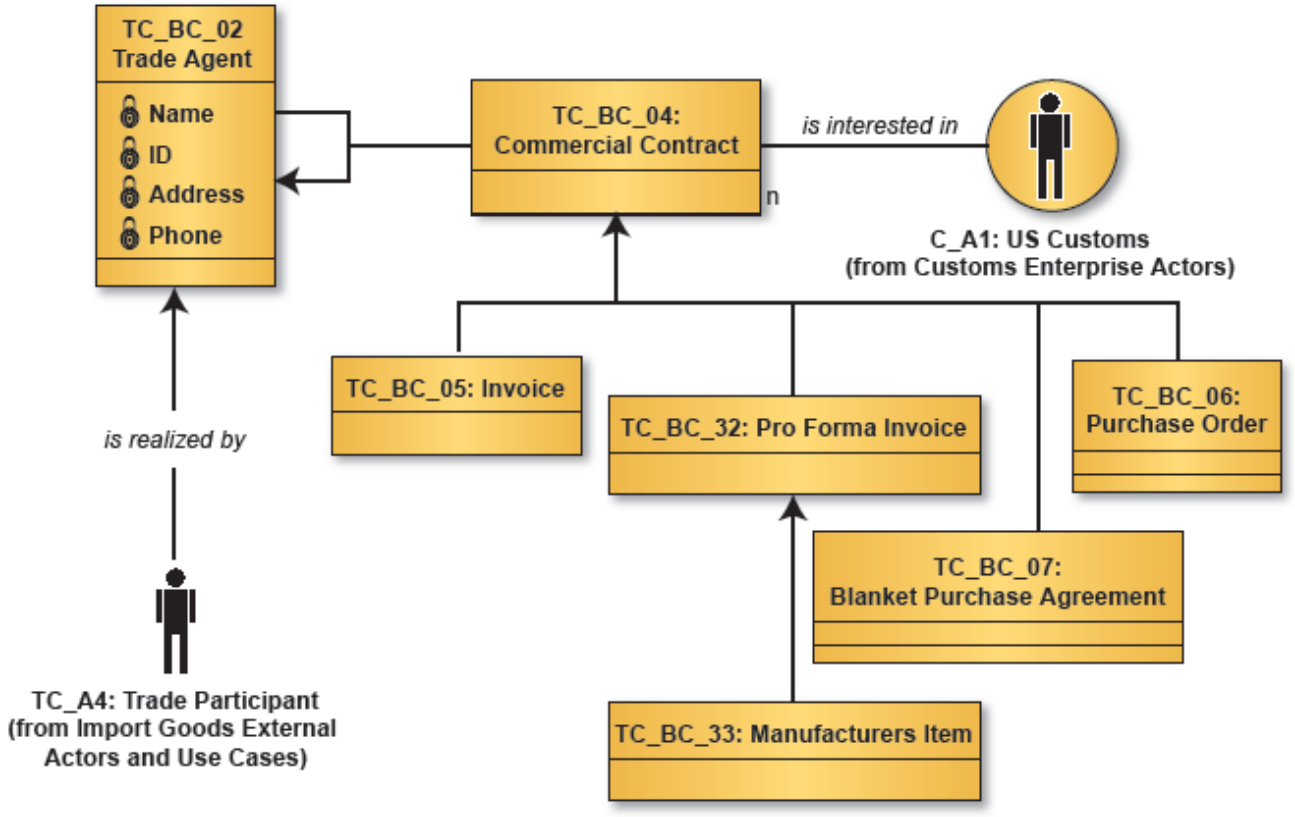


Figure 3: *UML Business Class Diagram, Trade Class Model (Commercial View)*[11]

All three types of model types above can be represented in Unified Modeling Language (UML), and a variety of tools exists for generating such models.

Certain industry sectors have modeling techniques specific to the sector concerned. For example, the Defense sector uses the following models:

- The **Node Connectivity Diagram** describes the business locations (nodes), the "needlines" between them, and the characteristics of the information exchanged. Node connectivity can be described at three levels: conceptual, logical, and physical. Each needline indicates the need for some kind of information transfer between the two connected nodes. A node can represent a role (e.g., a CIO); an organizational unit; a business location or facility; and so on. An arrow indicating the direction of information flow is annotated to describe the characteristics of the data or information; for example, its content; media; security, or classification level; timeliness; and requirements for information system interoperability.

  This can be represented using an Information Flow diagram in UML.

- The **Information Exchange Matrix** documents the information exchange requirements for an enterprise architecture. Information exchange requirements express the relationships across three basic entities (activities, business nodes and their elements, and information flow), and focus on characteristics of the information exchange, such as performance and security. They identify who exchanges what information with whom, why the information is necessary, and in what manner.

[11] Taken from "A Practical Guide to Federal Enterprise Architecture", CIO Council, February 2001.
[Note: Not a good example of UML.]

Although originally developed for use in the Defense sector, these models are finding increasing use in the other sectors of government, and may also be considered for use in non-government environments.

Enterprise Continuum

As part of Phase B, the architecture team will need to consider what relevant Business Architecture resources are available from the Enterprise Continuum: in particular:

- Generic business models relevant to the organization's industry sector. (These are "Industry Architectures", in terms of the Enterprise Continuum.) For example:
  - The Object Management Group (OMG) has a number of vertical Domain Task Forces developing business models relevant to specific vertical domains such as Healthcare, Transportation, Finance, etc.
  - The TeleMangement Forum (TMF) has developed detailed business models relevant to the Telecommunications industry.

    There is joint work with OMG to align with MDA standards.
  - Government departments and agencies in different countries have reference models and frameworks mandated for use, intended to promote cross-departmental integration and interoperability. An example is the Federal Enterprise Architecture Business Reference Model, which is a function-driven framework for describing the business operations of the Federal Government independent of the agencies that perform them.

    These reference frameworks can be represented as Classification Schemas in RAS, allowing any models to be categorized against the relevant parts of any reference framework in use.
- Business models relevant to common high-level business domains – such as electronic commerce, supply chain management, etc. – that are published within the IT industry. (These are "Common Systems Architectures", in terms of the Enterprise Continuum.) For example:
  - The Resource-Event-Agent (REA) business model was originally created by William E. McCarthy of Michigan State University, mainly for modeling of accounting systems. It has proved so useful for better understanding of business processes that it has become one of the major modeling frameworks for both traditional enterprises and e-Commerce systems. In particular, it has been extended by Robert Haugen and McCarthy for supply chain management.
  - The STEP Framework (STandard for the Exchange of Product model data) is concerned with product design and supply chain interworking. STEP is an ISO standard (ISO 10303). Implementation of the STEP standard has been led by some large aerospace manufacturers, and has also been taken up in other industries that have a need for complex graphic and process data, such as the construction industry.
  - The Open Applications Group (OAG) is focused on defining a framework for allowing heterogeneous business applications to communicate together. Its OAGIS integration model and specification address the needs of traditional Enterprise Resource Planning (ERP) integration, as well as supply chain management and electronic commerce.
  - RosettaNet is a consortium created by leading companies in the computer, electronic component, and semiconductor manufacturing supply chains. Its mission is to develop a complete set of standard e-Business processes for these supply chains, and to promote and support their adoption and use.
- Enterprise-specific building blocks (process components, business rules, job descriptions, etc.)

Gap Analysis

The Business Motivation Metamodel (BMM) allows for the capture of assessments and gaps; for example, as the result of applying a SWOT (Strengths Weaknesses Opportunities Threats) technique.

A key step in validating an architecture is to consider what may have been forgotten. The architecture must support all of the essential information processing needs of the organization. The most critical source of gaps that should be considered is stakeholder concerns that have not been addressed in prior architectural work.

Other potential sources of gaps:

- People gaps (e.g., cross-training requirements)
- Process gaps (e.g., process inefficiencies)
- Tools gaps (e.g., duplicate or missing tool functionality)
- Information gaps
- Measurement gaps
- Financial gaps
- Facilities gaps (buildings, office space, etc.)

Gap analysis highlights services and/or functions that have been accidentally left out, deliberately eliminated, or are yet to be developed or procured. Figure 5 illustrates an example of a gap analysis matrix. The suggested steps are as follows:

- Draw up a matrix with all the Business Architecture building blocks of the current architecture on the vertical axis, and all the Business Architecture building blocks of the Target Business Architecture on the horizontal axis. In creating the matrix it is imperative to use terminology that is accurate and consistent.
- Add to the Current Architecture axis a final row labeled “New Business Architecture Building Blocks”, and to the Target Architecture axis a final column labeled “Eliminated Business Architecture Building Blocks”.
- Where a Business Architecture building block is available in both the current and Target Architectures, record this with “Included” at the intersecting cell.
- Where a Business Architecture building block from the current architecture is missing in the Target Architecture, each must be reviewed. If it was correctly eliminated, mark it as such in the appropriate “Eliminated” cell. If it was not, you have uncovered an accidental omission in your new architecture that must be addressed by reinstating the Business Architecture building block in the next iteration of the architecture design – mark it as such in the appropriate “Eliminated” cell.
- Where a Business Architecture building block from the Target Architecture cannot be found in the current architecture, mark it at the intersection with the “New” row, as a gap that needs to filled, either by developing or procuring the building block.

When the exercise is complete, anything under “Eliminated Services” or “New Services” is a gap, which should either be explained as correctly eliminated, or marked as to be addressed by reinstating or developing/procuring the function.

Inputs

The inputs to Phase B are:

- Request for Architecture Work
- Approved Statement of Architecture Work/Project Definition
- Refined statements of business principles, business goals, and strategic drivers
- Architecture principles
- Enterprise Continuum
- Architecture Vision/Business Scenario, including:
  - Baseline Business Architecture, Version 1
  - Baseline Technology Architecture, Version 1
  - Target Business Architecture, Version 1
  - Target Technology Architecture, Version 1

Steps

The level of detail addressed in Phase B will depend on the scope and goals of the overall architecture effort.

New business processes being introduced as part of this effort will need to be defined in detail during Phase B. Existing business processes to be carried over and supported in the target environment may already have been adequately defined in previous architectural work; but, if not, they too will need to be defined in Phase B.

Key steps in Phase B include the following:

Note: The order of the following steps should be adapted to the situation at hand: in particular, determine whether in this situation it is appropriate to do Baseline Description or Target Architecture development first, as described in Part II: ADM, Introduction to the ADM.

1. **Business Architecture Baseline Description**. Develop a Baseline Description of the existing Business Architecture, to the extent necessary to support the Target Business Architecture. The scope and level of detail to be defined will depend on the extent to which existing business elements are likely to be carried over into the Target Business Architecture, and on whether existing architectural descriptions exist, as described under Approach, above. To the extent possible, identify the relevant Business Architecture building blocks, drawing on the Architecture Continuum.

   MDA standards: Business Motivation Metamodel (BMM); Business Process Definition Metamodel (BPDM); Organization Structure Metamodel (OSM); Business Semantics of Business Rules (BSBR)

2. **Reference Models, Viewpoints, and Tools:**
   - **Select relevant Business Architecture resources** (reference models, patterns, etc.) from the Architecture Continuum, on the basis of the business drivers, and the stakeholders and concerns.
   - **Select relevant Business Architecture viewpoints** (e.g., Operations; Management; Financial); i.e., those that will enable the architect to demonstrate how the stakeholder concerns are being addressed in the Business Architecture.

- **Identify appropriate tools and techniques** to be used for capture, modeling, and analysis, in association with the selected viewpoints. Depending on the degree of sophistication warranted, these may comprise simple documents or spreadsheets, or more sophisticated modeling tools and techniques such as activity models, business process models, use-case models, etc.

3. **Architecture Model(s):**
   - For each viewpoint, create the model for the specific view required, using the selected tool or method.
   - Assure that all stakeholder concerns are covered. If they are not, create new models to address concerns not covered, or augment existing models (see above). Business scenarios are a useful technique to discover and document business requirements, and may be used iteratively, at different levels of detail in the hierarchical decomposition of the Business Architecture. Business scenarios are described in Part IV: Resource Base, Business Scenarios. Other techniques may be used, if required. Create models of the following:
     - **Organization Structure**: Document the organization structure, identifying business locations and relating them to organizational units.

       MDA standard: Organization Structure Metamodel (OSM)
     - **Business Goals and Objectives**: Document business goals and objectives for each organizational unit.

       MDA standard: Business Motivation Metamodel (BMM)
     - **Business Functions**: Identify and define business functions. This is a detailed, recursive step involving successive decomposition of major functional areas into sub-functions.

       MDA standard: Business Process Definition Metamodel (BPDM)
     - **Business Services**: The services that each enterprise unit provides to its customers, both internally and externally.

       MDA standard: Business Process Definition Metamodel (BPDM)
     - **Business Processes**: I*n*cluding measures and deliverables.

       MDA standard: Business Process Definition Metamodel (BPDM)
     - **Business Roles**: Including development and modification of skills requirements.

       MDA standard: Enterprise Organization Metamodel (EOM)
     - **Correlation of Organization and Functions**: Relate business functions to organizational units in the form of a matrix report.

       MDA standards: IT Portfolio Management Facility (ITPMF); and/or references from Business Process Definition Metamodel (BPDM) to Organization Structure Metamodel (OSM) and Business Process Definition Metamodel (BPDM)
   - **Information requirements.** Identify for each business function: when, where, how often, and by whom the function is performed; what information is used to perform it, and its source(s); and what opportunities exist for improvements. Include information that needs to be created, retrieved, updated, and deleted. The level of detail to be defined will depend on the scope and focus of the current architecture effort, as describe under Approach, above. Focus on what will be worthwhile collecting for the purpose at hand.
   - Perform trade-off analysis to resolve conflicts (if any) among the different views.

     One method of doing this is CMU/SEI's Architecture Trade-off Analysis (ATA) Method.
   - Validate that the models support the principles, objectives, and constraints.
   - Note changes to the viewpoint represented in the selected models from the Architecture Continuum, and document.

- Test architecture models for completeness against requirements.

4. **Select Business Architecture building blocks** (e.g., business services)
   - Identify required building blocks and check against existing library of building blocks, re-using as appropriate.
   - Where necessary, define new Business Architecture building blocks.
5. **Conduct a formal checkpoint review** of the architecture model and building blocks with stakeholders.
6. **Review non-functional (qualitative) criteria** (e.g., performance, costs, volumes). Use to specify required service levels; for example, via formal Service Level Agreements (SLAs).
7. **Complete the Business Architecture**:
   - Select standards for each of the Architectural Building Blocks, re-using as much as possible from the reference models selected from the Architecture Continuum.
   - Fully document each Architectural Building Block.
   - Final cross-check of overall architecture against business goals. Document rationale for building block decisions in architecture document.
   - Document final requirements traceability report.
   - Document final mapping of the architecture within the Architecture Continuum. From the selected Architectural Building Blocks, identify those that might be re-used (working practices, roles, business relationships, job descriptions, etc.), and publish via the architecture repository.
   - Document rationale for building block decisions in architecture document.
   - Prepare Business Architecture report. Generate the Business Architecture document, comprising some or all of:
     - A business footprint (a high-level description of the people and locations involved with key business functions)
     - A detailed description of business functions and their information needs
     - A management footprint (showing span of control and accountability)
     - Standards, rules, and guidelines showing working practices, legislation, financial measures, etc.
     - A skills matrix and set of job descriptions

     If appropriate, use reports and/or graphics generated by modeling tools to demonstrate key views of the architecture. Route the Business Architecture document for review by relevant stakeholders, and incorporate feedback.
   - **Checkpoint**: Check the original motivation for the architecture project and the Statement of Architecture Work against the proposed Business Architecture, asking if it is fit for the purpose of supporting subsequent work in the other architecture domains. Refine the proposed Business Architecture only if necessary.
8. **Gap analysis and report:**
   - Create gap matrix.
   - Identify building blocks to be carried over, classifying as either changed or unchanged.
   - Identify eliminated building blocks.
   - Identify new building blocks.
   - Identify gaps and classify as those that should be developed and those that should be procured.

Outputs

The outputs of Phase B are:

- Statement of Architecture Work (updated if necessary)
- Validated business principles, business goals, and strategic drivers

  MDA standard: Business Motivation Metamodel (BMM)
- Target Business Architecture, Version 2 (detailed), including:
  - **Organization Structure**: Identifying business locations and relating them to organizational units.

    MDA standard: Organization Structure Metamodel (OSM)
  - **Business Goals and Objectives**: For each organizational unit.

    MDA standard: Business Motivation Metamodel (BMM)
  - **Business Functions**: A detailed, recursive step involving successive decomposition of major functional areas into sub-functions.

    MDA standard: Business Process Definition Metamodel (BPDM)
  - **Business Service**s: The services that each enterprise unit provides to its customers, both internally and externally.

    MDA standard: Business Process Definition Metamodel (BPDM)
  - **Business Processes**: Including measures and deliverables.

    MDA standard: Business Process Definition Metamodel (BPDM)
  - **Business Roles**: Including development and modification of skills requirements.

    MDA standard: Enterprise Organization Metamodel (EOM)
  - **Correlation of Organization and Functions**: Relate business functions to organizational units in the form of a matrix report.

    MDA standards: IT Portfolio Management Facility (ITPMF); and/or references from Business Process Definition Metamodel (BPDM) to Organization Structure Metamodel (OSM) and Business Process Definition Metamodel (BPDM)
- Baseline Business Architecture, Version 2 (detailed), if appropriate

  MDA standards: UML (Use-Cases); Business Motivation Metamodel (BMM); Business Process Definition Metamodel (BPDM); Organization Structure Metamodel (OSM); Business Semantics of Business Rules (BSBR)
- Views corresponding to the selected viewpoints addressing key stakeholder concerns

  MDA standards: UML (Use-Cases); Business Motivation Metamodel (BMM); Business Process Definition Metamodel (BPDM); Organization Structure Metamodel (OSM); Business Semantics of Business Rules (BSBR)
- Gap analysis results
- Technical requirements (drivers for the Technology Architecture work) – identifying, categorizing, and prioritizing the implications for work in the remaining architecture domains; for example, by a dependency/priority matrix. (For example, guiding trade-off between speed of transaction processing and security.) List the specific models that are expected to be produced (for example, expressed as primitives of the Zachman Framework).

  MDA standards: UML (Use-Cases); SysML

- Business Architecture report
- Updated business requirements

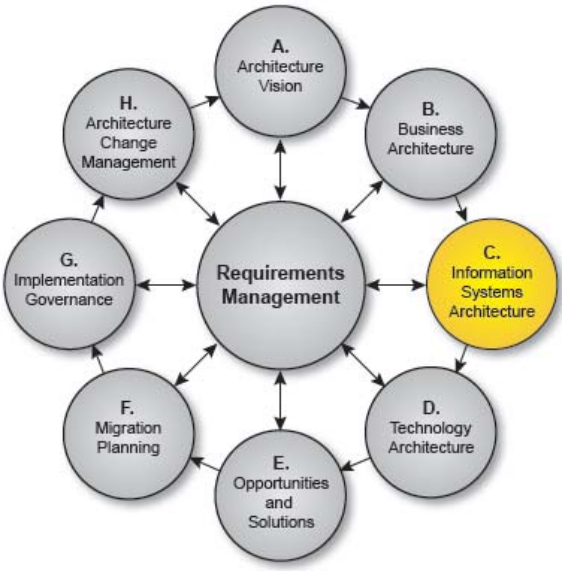
  MDA standard: SysML

Phase C: Information System Architectures

Objective

The objective of Phase C is to develop Target Architectures covering either or both (depending on project scope) of the Data and Application Systems domains.

The scope of the business processes supported in Phase C is limited to those that are supported by IT, and the interfaces of those IT-related processes to non-IT-related processes.



Development

Phase C involves some combination of Data and Applications Architecture, in either order. Advocates exist for both sequences. For example, Spewak's Enterprise Architecture Planning (EAP) recommends a data-driven approach.

On the other hand, major applications systems – such as those for Enterprise Resource Planning (ERP), customer relationship management, etc. – often provide a combination of technology infrastructure and business application logic, and some organizations take an application-driven approach, whereby they recognize certain key applications as forming the core underpinning of the mission-critical business processes, and take the implementation and integration of those core applications as the primary focus of architecture effort (the integration issues often constituting a major challenge).

Implementation

Implementation of these architectures may not necessarily follow the same order. For example, one common implementation approach is top-down design and bottom-up implementation:

- Design:
  - Business Architecture design
  - Data (or Applications) Architecture design
  - Applications (or Data) Architecture design
  - Technology Architecture design
- Implementation:
  - Technology Architecture implementation
  - Applications (or Data) Architecture implementation
  - Data (or Applications) Architecture implementation
  - Business Architecture implementation

An alternative approach is a data-driven sequence, whereby application systems that create data are implemented first, then applications that process the data, and finally applications that archive data.

Inputs

Inputs to Phase C are:

- Applications principles (if existing)
- Data principles (if existing)
- Request for Architecture Work
- Statement of Architecture Work
- Architecture Vision
- Enterprise Continuum
- Baseline Business Architecture, Version 2 (detailed), if appropriate
- Target Business Architecture, Version 2 (detailed)
- Relevant technical requirements that will apply to Phase C
- Gap analysis (from Business Architecture)
- Re-usable building blocks (from organization's Architecture Continuum, if available)

Steps

Detailed steps for Phase C are given separately for each architecture domain:

- Data Architecture
- Applications Architecture

Outputs

The main outputs are as follows:

- Statement of Architecture Work (updated if necessary)
- Target Data Architecture
    - Conceptual data model

      MDA standards: Common Warehouse Metamodel (CWM); Ontology Definition Metamodel (ODM)
    - Logical data model

      MDA standard: Common Warehouse Metamodel (CWM)
    - Data management process models

      MDA standard: Common Warehouse Metamodel (CWM)
    - Data entity/business function matrix

      MDA standards: (generic) references from Business Process Definition Metamodel (BPDM) to Common Warehouse Metamodel (CWM) or IT Portfolio Management Facility (ITPMF)
    - Data interoperability requirements

      MDA standards: Common Warehouse Metamodel (CWM); Enterprise Distributed Object Computing (EDOC)
- Target Applications Architecture
    - Process systems model

MDA standards: IT Portfolio Management Facility (ITPMF); UML (Activity Diagrams); Enterprise Distributed Object Computing (EDOC)

- Place systems model

  MDA standard: IT Portfolio Management Facility (ITPMF)

- Time systems model

  MDA standard: UML (Sequence Diagrams)

- People systems model

  MDA standard: IT Portfolio Management Facility (ITPMF)

- Applications interoperability requirements

  MDA standard: UML (Component Diagrams) or Enterprise Distributed Object Computing (EDOC)

- Data Architecture views corresponding to the selected viewpoints addressing key stakeholder concerns; e.g.:
  - Data Dissemination view

    MDA standard: IT Portfolio Management Facility (ITPMF)

  - Data Lifecycle view

    MDA standard: UML State diagrams

  - Data Security view

    MDA standard: IT Portfolio Management Facility (ITPMF)

  - Data Model Management view

    MDA standard: (generic) Meta Object Facility (MOF)

- Applications Architecture views corresponding to the selected viewpoints addressing key stakeholder concerns; e.g.:
  - Common Applications Services view

    MDA standards: IT Portfolio Management Facility (ITPMF); UML (Component Diagrams) or Enterprise Distributed Object Computing (EDOC)

  - Applications Interoperability view

    MDA standard: UML (Component Diagrams) or Enterprise Distributed Object Computing (EDOC)

  - Applications/Information view

    MDA standards: IT Portfolio Management Facility (ITPMF); UML (Component Diagrams) or Enterprise Distributed Object Computing (EDOC)

  - Applications/User Locations view

    MDA standard: IT Portfolio Management Facility (ITPMF)

- Data Architecture report, summarizing what was done and the key findings
- Applications Architecture report, summarizing what was done and the key findings
- Gap analysis:
  - Areas where the Business Architecture may need to change to cater for changes in the Data and/or Applications Architecture
  - Constraints on the Technology Architecture about to be designed

- Impact Analysis

  No mapping, but views on other metamodels can be defined using MOF QVT.
- Updated business requirements (if appropriate)

  MDA standard: SysML

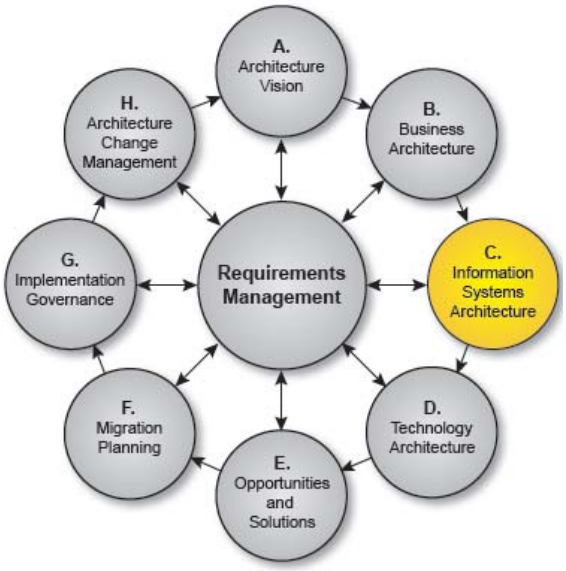
Phase C: Information System Architectures: Data Architecture

Objective

The objective here is to define the major types and sources of data necessary to support the business, in a way that is:

- Understandable by stakeholders
- Complete and consistent
- Stable

It is important to note that this effort is *not* concerned with database design. The goal is to define the data entities relevant to the enterprise, not to design logical or physical storage systems. (However, linkages to existing files and databases may be developed, and may demonstrate significant areas for improvement.)



Enterprise Continuum

As part of this phase, the architecture team will need to consider what relevant Data Architecture resources are available in the organization's Enterprise Continuum; in particular, generic data models relevant to the organization's industry "vertical" sector. For example:

- ARTS has defined a data model for the Retail industry.
- POSC has defined a data model for the Petrotechnical industry.

Gap Analysis

A key step in validating an architecture is to consider what may have been forgotten. The architecture must support all of the essential information processing needs of the organization. The most critical source of gaps that should be considered is stakeholder concerns that have not been addressed in architectural work.

Types of data gap:

- Data not located where it is needed
- Not the data that is needed
- Data not available when needed
- Data not created
- Data not consumed
- Data relationship gaps, etc.

Gap analysis highlights shortfalls in data services and/or data elements that have been accidentally left out, deliberately eliminated, or are yet to be defined. Figure 5 illustrates an example of a gap analysis matrix. The suggested steps are as follows:

- Draw up a matrix with all the Data Architecture building blocks of the current architecture on

the vertical axis, and all the Data Architecture building blocks of the Target Data Architecture on the horizontal axis. In creating the matrix, it is imperative to use terminology that is accurate and consistent.

- Add to the Current Architecture axis a final row labeled “New Data Architecture Building Blocks”, and to the Target Architecture axis a final column labeled “Eliminated Data Architecture Building Blocks”.
- Where a Data Architecture building block is available in both the current and Target Architectures, record this with “Included” at the intersecting cell.
- Where a Data Architecture building block from the current architecture is missing in the Target Architecture, each must be reviewed. If it was correctly eliminated, mark it as such in the appropriate “Eliminated” cell. If it was not, you have uncovered an accidental omission in your new architecture that must be addressed by reinstating the Data Architecture building block in the next iteration of the architecture design – mark it as such in the appropriate “Eliminated” cell.
- Where a Data Architecture building block from the Target Architecture cannot be found in the current architecture, mark it at the intersection with the “New” row, as a gap that needs to filled, either by defining or inheriting the building block.

When the exercise is complete, anything under “Eliminated Services” or “New Services” is a gap, which should either be explained as correctly eliminated, or marked as to be addressed by reinstating or developing/procuring the function.

The Reusable Asset Specification (RAS) can be used to capture the building blocks and their status.

Inputs

Inputs to this phase are:

- Data principles (if existing)

  MDA standards: Could be represented by Policy elements in Business Motivation Metamodel (BMM), but more likely in Business Semantics of Business Rules (BSBR) or Ontology Definition Metamodel (ODM). A more generic option is to use UML Class diagrams with attached constraints (in OCL or natural language).
- Request for Architecture Work
- Statement of Architecture Work
- Architecture Vision
- Relevant technical requirements that will apply to this phase
- Gap analysis (from Business Architecture)
- Baseline Business Architecture, Version 2 (detailed), if appropriate
- Target Business Architecture, Version 2 (detailed)
- Re-usable building blocks (from organization's Enterprise Continuum, if available), in particular, definitions of current data

Steps

1. **Data Architecture Baseline Description.** Develop a Baseline Description of the existing Data Architecture, to the extent necessary to support the Target Data Architecture. The scope and

level of detail to be defined will depend on the extent to which existing data elements are likely to be carried over into the Target Data Architecture, and on whether existing architectural descriptions exist, as described under Approach. To the extent possible, identify the relevant Data Architecture building blocks, drawing on the Architecture Continuum, and review/verify the following primitives from the Zachman Framework:

- Conceptual data model (entities, attributes, and relationships)
  - Entity-relationship diagrams illustrating views of the Data Architecture to address the concerns of stakeholders.

    MDA standards: Common Warehouse Metamodel (CWM); Ontology Definition Metamodel (ODM); Software Process Engineering Metamodel (SPEM) can also be used for Data Management Process Models.
  - Logical data model (logical views of the actual data of interest)

    MDA standard: Common Warehouse Metamodel (CWM) or UML (class diagrams)
  - Data management process models, including data dissemination view, data lifecycle view, data security view, and data model management view

    MDA standard: Common Warehouse Metamodel (CWM) – addresses dissemination and lifecycle also via Transformation and Deployment models.
  - Data entity/business function matrix in the Business Architecture

    MDA standards: (generic) references from Business Process Definition Metamodel (BPDM) to Common Warehouse Metamodel (CWM) or IT Portfolio Management Facility (ITPMF)

2. **Principles, Reference Models, Viewpoints, and Tools:**

- Review and validate (or generate, if necessary) the set of data principles.

  These will normally form part of an overarching set of architecture principles. Guidelines for developing and applying principles, and a sample set of data principles, are given in Part IV: Resource Base, Architecture Principles.
- Select relevant Data Architecture resources (reference models, patterns, etc.) from the Architecture Continuum, on the basis of the business drivers, and the stakeholders and concerns.

  MDA standard: Reusable Asset Specification (RAS)
- Select relevant Data Architecture viewpoints (for example, stakeholders of the data – regulatory bodies, users, generators, subjects, auditors, etc.; various time dimensions – real-time, reporting period, event-driven, etc.; locations; business processes); i.e., those that will enable the architect to demonstrate how the stakeholder concerns are being addressed in the Data Architecture.
- Identify appropriate tools and techniques to be used for data capture, modeling, and analysis, in association with the selected viewpoints. Depending on the degree of sophistication warranted, these may comprise simple documents or spreadsheets, or more sophisticated modeling tools and techniques such as data management models, data models, etc. Examples of data modeling techniques are:
  - IDEF
  - Object Role Modeling
  - UML

  MDA standards: Support for MDA can help guide what is "appropriate" for tools.

3. **Architecture Model(s):**
   - For each viewpoint, create the model for the specific view required, using the selected tool or method.

     Examples of logical data models are:

     - The C4ISR Architecture Framework Logical Data Model
     - ARTS Data Model for the Retail industry
     - POSC Data Model for the Petrotechnical industry
   - Assure that all stakeholder concerns are covered. If they are not, create new models to address concerns not covered, or augment existing models (see above). Model the following:
     - Conceptual data model (entities, attributes, and relationships)

       Draw entity-relationship diagrams to illustrate views of the Data Architecture to address the concerns of stakeholders.

       MDA standards: Common Warehouse Metamodel (CWM); Ontology Definition Metamodel (ODM), UML
     - Logical data model (logical views of the actual data of interest)

       MDA standard: Common Warehouse Metamodel (CWM), UML
     - Data management process models, including data dissemination view, data lifecycle view, data security view, and data model management view

       MDA standard: Common Warehouse Metamodel (CWM)
   - Relate data entities to business functions in the Business Architecture, indicating which of the CRUD operations (Create, Reference, Update, and Delete) are performed by which functions.
     - Relate each lowest-level business function in the Business Architecture to the set of data entities, indicating which of the CRUD operations (Create, Reference, Update, and Delete) are performed by the function concerned.
     - Generate entity-business function matrices tabulating all the relationships.
     - Review and validate the entity-business function matrices, checking that each entity is created by at least one function, and referenced or updated by at least one other function.
     - Time permitting, relate entities to the application systems described in the Baseline Description.

       MDA standards: (generic) references from Business Process Definition Metamodel (BPDM) to Common Warehouse Metamodel (CWM) or IT Portfolio Management Facility (ITPMF)
   - Ensure that all information requirements in the Business Architecture are met.
   - Perform trade-off analysis to resolve conflicts (if any) among the different views.

     One method of doing this is CMU/SEI's Architecture Trade-off Analysis (ATA) Method.
   - Validate that the models support the principles, objectives, and constraints.
   - Note changes to the viewpoint represented in the selected models from the Architecture Continuum, and document.
   - Test architecture models for completeness against requirements.
4. **Select Data Architecture building blocks** (e.g., metamodels)
   - Identify required building blocks and check against existing library of building blocks, re-using as appropriate.

- Where necessary, define new Data Architecture building blocks.

  MDA standard: Reusable Asset Specification (RAS)

5. **Conduct a formal checkpoint review** of the architecture model and building blocks with stakeholders.
   - Review the entity-business function matrices generated in Step 3, and the Business Architecture generated in Phase B.
6. **Review the qualitative criteria** (e.g., security, availability, accuracy, performance, costs, volumes), providing as many measurable criteria as possible (e.g., privacy/confidentiality, reliability, minimum tolerable data losses, maximum data volumes at peak times, etc.). Use to specify required service levels for data services; for example, via formal Service Level Agreements (SLAs).
   - The goal here is to guide the Applications and Technology Architecture efforts as to the qualities required in the applications, and the underlying technology, that manage and process the data.
7. **Complete the Data Architecture**:
   - Select standards for each of the Architectural Building Blocks, re-using as much as possible from the reference models selected from the Architecture Continuum.
   - Fully document each Architectural Building Block.

     MDA standard: Reusable Asset Specification (RAS)
   - Final cross-check of overall architecture against business requirements. Document rationale for building block decisions in architecture document.
   - Document final requirements traceability report.
   - Document final mapping of the architecture within the Architecture Continuum. From the selected Architectural Building Blocks, identify those that might be re-used, and publish via the architecture repository.
   - Document rationale for building block decisions in architecture document.
   - Prepare Data Architecture report. Generate the Data Architecture document, comprising some or all of:
     - Conceptual data model
     - Logical data model
     - Data management process model
     - Data entity/business function matrix
     - Data interoperability requirements (e.g., XML schema, security policies)

     MDA standards for data interoperability requirements: Common Warehouse Metamodel (CWM); Enterprise Distributed Object Computing (EDOC)

     If appropriate, use reports and/or graphics generated by modeling tools to demonstrate key views of the architecture. Route the Data Architecture document for review by relevant stakeholders, and incorporate feedback.
   - Checkpoint/Impact analysis: Check the original motivation for the architecture project and the Statement of Architecture Work against the proposed Data Architecture**.** Conduct an impact analysis to:
     - Identify any areas where the Business Architecture (e.g., business practices) may need to change to cater for changes in the Data Architecture (for example, changes to forms or procedures, application systems, or database systems).

If the impact is significant, this may warrant the Business Architecture being revisited.

- Identify any areas where the Applications Architecture (if generated at this point) may need to change to cater for changes in the Data Architecture (or to identify constraints on the Applications Architecture about to be designed).

  If the impact is significant, this may warrant the Applications Architecture being revisited, if already developed in this cycle.
- Identify any constraints on the Technology Architecture about to be designed.
- Refine the proposed Data Architecture only if necessary.

8. **Gap analysis and report**:
   - Create gap matrix.
   - Identify building blocks to be carried over, classifying as either changed or unchanged.
   - Identify eliminated building blocks.
   - Identify new building blocks.
   - Identify gaps and classify as those that should be defined and those inherited.

Outputs

The outputs of this phase are:

- Statement of Architecture Work (updated if necessary)
- Data Architecture Baseline Description, if appropriate

  MDA standard: Common Warehouse Metamodel (CWM)
- Validated data principles, or new data principles (if generated here)

  Could be represented by Policy elements in Business Motivation Metamodel (BMM), but more likely in Business Semantics of Business Rules (BSBR) or Ontology Definition Metamodel (ODM). A more generic option is to use UML Class diagrams with attached constraints (in OCL or natural language).
- Target Data Architecture
  - Conceptual data model

    MDA standards: Common Warehouse Metamodel (CWM); Ontology Definition Metamodel (ODM)
  - Logical data model

    MDA standard: Common Warehouse Metamodel (CWM)
  - Data management process models

    MDA standard: Common Warehouse Metamodel (CWM)
  - Data entity/business function matrix

    MDA standards: (generic) references from Business Process Definition Metamodel (BPDM) to Common Warehouse Metamodel (CWM) or IT Portfolio Management Facility (ITPMF)
  - Data interoperability requirements

    MDA standards: Common Warehouse Metamodel (CWM); Enterprise Distributed Object Computing (EDOC)

- Viewpoints addressing key stakeholder concerns

  No mapping, but views on other metamodels.
- Views corresponding to the selected viewpoints; e.g.:
  - Data Dissemination view

    MDA standard: IT Portfolio Management Facility (ITPMF)
  - Data Lifecycle view

    MDA standard: UML State diagrams
  - Data Security view

    MDA standard: IT Portfolio Management Facility (ITPMF)
  - Data Model Management view

    MDA standard: (generic) Meta Object Facility (MOF)
- Gap analysis results
- Relevant technical requirements that will apply to this evolution of the architecture development cycle

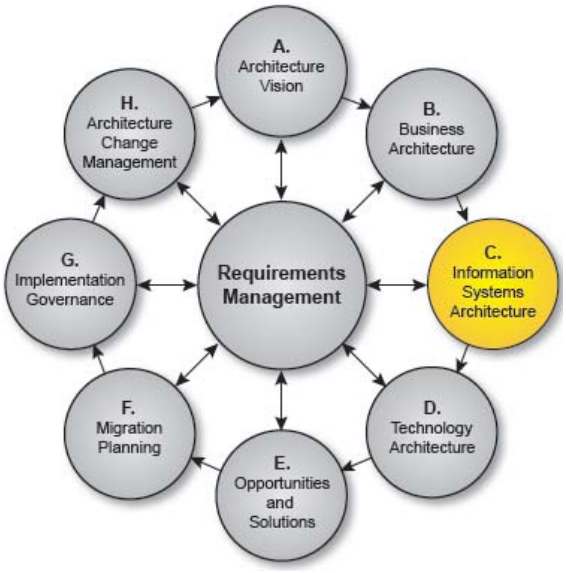
  MDA standard: IT Portfolio Management Facility (ITPMF)
- Data Architecture report, summarizing what was done and the key findings
- Impact Analysis
  - Areas where the Business Architecture may need to change to cater for changes in the Data Architecture
  - Identify any areas where the Applications Architecture (if generated at this point) may need to change to cater for changes in the Data Architecture
  - Constraints on the Technology Architecture about to be designed

    No mapping, but views on other metamodels can be defined using MOF QVT.
- Updated business requirements (if appropriate)

  MDA standard: SysML

Phase C: Information System Architectures: Applications Architecture

Objective



The objective here is to define the major kinds of application system necessary to process the data and support the business.

It is important to note that this effort is *not* concerned with applications systems design. The goal is to define what kinds of application systems are relevant to the enterprise, and what those applications need to do in order to manage data and to present information to the human and computer actors in the enterprise.

The applications are not described as computer systems, but as logical groups of capabilities that manage the data objects in the Data Architecture and support the business functions in the Business Architecture. The applications and their capabilities are defined without reference to particular technologies. The applications are stable and relatively unchanging over time, whereas the technology used to implement them will change over time, based on the technologies currently available and changing business needs.

Enterprise Continuum

As part of this phase, the architecture team will need to consider what relevant Applications Architecture resources are available in the Enterprise Continuum.

In particular:

- Generic business models relevant to your organization's industry "vertical" sector; for example:
  - The TeleManagement Forum (TMF) has developed detailed applications models relevant to the Telecommunications industry.
  - The Object Management Group (OMG) has a number of vertical Domain Task Forces developing software models relevant to specific domains such as Healthcare, Transportation, Finance, etc.
- Application models relevant to common high-level business functions, such as electronic commerce, supply chain management, etc.

The Open Group has a Reference Model for Integrated Information Infrastructure (see Part III: Enterprise Continuum, III-RM) that focuses on the application-level components and services necessary to provide an integrated information infrastructure.

In addition, the ebXML Initiative aims to provide an open, XML-based infrastructure enabling global use of electronic business information in an interoperable, secure, and consistent manner. UML is used for modeling aspects and XML for syntax aspects. The initiative was formed as a joint venture by the UN/CEFACT community and the OASIS Consortium, with ANSI X.12 also fully participating.

Gap Analysis

A key step in validating an architecture is to consider what may have been forgotten. The architecture must support all of the essential information processing needs of the organization. The most critical source of gaps that should be considered is stakeholder concerns that have not been addressed in architectural work.

Gap analysis highlights shortfalls in applications services and/or applications components that have been accidentally left out, deliberately eliminated, or are yet to be defined. Figure 5 illustrates an example of a gap analysis matrix. The suggested steps are as follows:

1. Draw up a matrix with all the applications in the current Applications Architecture on the vertical axis, and all the applications in the Target Applications Architecture on the horizontal axis.
2. Add to the Current Architecture axis a final row labeled “New Applications”, and to the Target Architecture axis a final column labeled “Eliminated Applications”.
3. Where an application exists in both the current and Target Architectures, record this fact with “Retained” at the intersecting cell.
4. Where an application in the current architecture is missing in the Target Architecture, the current and Target Architectures must be reviewed.
    - If the current application was correctly eliminated, mark it as such in the appropriate “Eliminated Applications” cell.
    - If the current application is to be replaced, wholly or partly, by one or more applications in the target architecture, make a note to this effect in the corresponding intersecting cell(s).
    - If the current application was unintentionally eliminated in the Target Architecture, note this fact in the appropriate “Eliminated Applications” cell. The omission will need to be addressed in an iteration of the Target Applications Architecture design.
5. Where an application in the Target Applications Architecture cannot be found in the current architecture, mark it at the intersection with the “New” row, as a gap that needs to be filled, either by developing or procuring the application.
6. When the exercise is complete, anything under “Eliminated Applications” or “New Applications” is a potential gap, which should either be explained as correctly eliminated, or marked as to be addressed, either by reinstating in the Target Architecture, or by developing/procuring the application.
7. Check that the applications gap analysis is complete.

Inputs

Inputs to this phase are:

- Applications principles (if existing)

  Potentially represented by very general UML class diagrams.
- Request for Architecture Work
- Statement of Architecture Work
- Architecture Vision
- Relevant technical requirements that will apply to this phase
- Gap analysis (from Business Architecture)

- Baseline Business Architecture, Version 2 (detailed), if appropriate
- Target Business Architecture, Version 2 (detailed)
- Re-usable building blocks (from organization's Enterprise Continuum, if available)
- Data Architecture Baseline Description, if appropriate and if available
- Target Data Architecture, if available

Steps

1. **Applications Architecture Baseline Description.** Develop a Baseline Description of the existing Applications Architecture, to the extent necessary to support the Target Applications Architecture. The scope and level of detail to be defined will depend on the extent to which existing application components are likely to be carried over into the Target Applications Architecture, and on whether existing architectural descriptions exist, as described in Approach. Define for each application:
   - Name (short and long)
   - Who maintains
   - Owner(s)/business unit(s) responsible for requirements
   - Other users
   - Plain language description of what the application does (not how it does it)
   - Status (planned, operational, obsolete)
   - Business functions supported
   - Organizational units supported
   - Hardware/software platform(s) on which it runs
   - Networks used
   - Precedent and successor applications

   To the extent possible, identify the relevant Applications Architecture building blocks, drawing on the Architecture Continuum.
2. **Principles, Reference Models, Viewpoints, and Tools:**
   - Review and validate (or generate, if necessary) the set of application principles.

     These will normally form part of an overarching set of architecture principles. Guidelines for developing and applying principles, and a sample set of application principles, are given in Part IV: Resource Base, Architecture Principles.
   - Select relevant Applications Architecture resources (reference models, patterns, etc.) from the Architecture Continuum, on the basis of the business drivers, and the stakeholders and concerns.
   - Select relevant Applications Architecture viewpoints (for example, stakeholders of the applications – viewpoints relevant to functional and individual users of applications, Software Engineering view, Application-to-Application Communication view, Software Distribution view, Enterprise Manageability view, etc.) – i.e., those that will enable the architect to demonstrate how the stakeholder concerns are being addressed in the Applications Architecture.
   - Identify appropriate tools and techniques to be used for capture, modeling, and analysis, in association with the selected viewpoints. Depending on the degree of sophistication warranted, these may comprise simple documents or spreadsheets, or more sophisticated modeling tools and techniques.

Consider using platform-independent descriptions of business logic. For example, the OMG's Model-Driven Architecture (MDA) offers an approach to modeling Applications Architectures that preserves the business logic from changes to the underlying platform and implementation technology.

- *MDA standards: UML, Enterprise Distributed Object Computing (EDOC), or Business Semantics of Business Rules (BSBR)*
- *Support for MDA can help guide what is "appropriate" for tools.*

3. **Architecture Model(s)**:
   - For each viewpoint, create the model for the specific view required, using the selected tool or method. Examples of applications models are:
     - The TeleManagement Forum (TMF) has developed detailed applications models relevant to the Telecommunications industry.
     - The Object Management Group (OMG) has a number of vertical Domain Task Forces developing software models relevant to specific domains such as Healthcare, Transportation, Finance, etc.
   - Assure that all stakeholder concerns are covered. If they are not, create new models to address concerns not covered, or augment existing models (see above). Model at least the following:
     - Common Applications Services view – both those being consumed, and those being produced for others to consume.

       MDA standards: IT Portfolio Management Facility (ITPMF); UML (Component Diagrams) or Enterprise Distributed Object Computing (EDOC)
     - Applications Interoperability view, assumptions, dependencies, and standards (an example is the LISI Interoperability Model). Also, The Open Group has a Reference Model for Integrated Information Infrastructure (see Part III: Enterprise Continuum, III-RM) that can be used as a basis for this.

       MDA standard: UML (Component Diagrams) or Enterprise Distributed Object Computing (EDOC)
   - Relate the application systems to the business functions in the Business Architecture:
     - For each application, identify each lowest-level business function that it supports in the Business Architecture.
     - Generate application-business function matrices tabulating all the relationships.
     - Review and validate the application-business function matrices. In cases where a business function is supported by more than one application, check for any unnecessary duplication of functionality (and, if found, eliminate it by redefining the applications concerned).
     - Identify any business functions not supported by an application, and rationalize: either provide application support by amending or updating the set of application definitions, iterating Steps 1 to 3; or else document the reason why no application support is warranted.
     - Use the application-business function matrices generated above, and the business function-to-organizational-unit mappings (business functions cross-linked to the organizational units that perform them) contained in the Business Architecture, to relate the application systems to the organizational units that they support.
   - Ensure that all information requirements in the Business Architecture are met.
   - Perform trade-off analysis to resolve conflicts (if any) among the different views.

     One method of doing this is CMU/SEI's Architecture Trade-off Analysis (ATA) Method.

- Validate that the models support the principles, objectives, and constraints.
- Note changes to the viewpoint represented in the selected models from the Architecture Continuum, and document.
- Test architecture models for completeness against requirements.

4. **Identify candidate application systems:**
   - Review the re-usable Architectural Building Blocks and re-usable solution building blocks from the enterprise's Architecture Continuum, the business scenario description, and the Baseline Description, and list all the potential application systems.
   - If available, review the entity-to-business-function matrices from the Data Architecture, and identify potential applications to perform the required data management functions, and/or to automate particular business functions.
   - Even if a complete Data Architecture is not available, review whatever lists of data exist.
   - Develop a user location/applications matrix.
   - Brainstorm other potential application systems based on innovative use of new developments in technology.
   - Merge all lists into a single, de-duplicated list of candidate application systems, including for each a brief description of business function(s) supported and data/information managed.
   - Create application definitions for all candidate application systems. For each application:
     - Assign a unique name and identifier.
     - Write a brief description of what the application does (not how it works).
     - Write a brief description of the business benefits arising from the application.
     - Simplify complicated applications by decomposing them into two or more applications.
     - Ensure that the set of application definitions is internally consistent, by removing duplicate functionality as far as possible, and combining similar applications into one.
     - Identify technology requirements and candidate technology building blocks, where this affects the applications design, including re-usable solution building blocks from the Architecture Continuum, and external software packages.
     - Identify any critical infrastructure dependencies (e.g., operating system and network services required).
     - Identify any critical application dependencies, and minimize as far as possible.
     - Time permitting, relate the applications to the files and databases described in the Baseline Description, and/or to the data entities defined in the Data Architecture (if available).
     - Time permitting, draw simple diagrams to illustrate views of the Applications Architecture relevant to different stakeholders.
5. **Conduct a formal checkpoint review** of the architecture model and building blocks with stakeholders.
   - Review the application-business function matrices generated in Step 3, and the Business Architecture generated in Phase B.
6. **Review the qualitative criteria** (e.g., security, availability, performance, costs), providing as many measurable criteria as possible (e.g., privacy/confidentiality, reliability, minimum tolerable outages, cycle requirements and transaction volume requirements at peak and mean times, numbers and locations of users, etc.). Use to specify required service levels for Applications services; for example, via formal Service Level Agreements (SLAs).
   - The goal here is to guide the Data and Technology Architecture efforts as to the qualities

required in the data, and the underlying technology, that support and are processed by the application.

7. **Complete the Applications Architecture:**
    - Select standards for each of the Architectural Building Blocks, re-using as much as possible from the reference models selected from the Architecture Continuum.
    - Fully document each Architectural Building Block.
    - Final cross-check of overall architecture against business requirements. Document rationale for building block decisions in architecture document.
    - Document final requirements traceability report.
    - Document final mapping of the architecture within the Architecture Continuum. From the selected Architectural Building Blocks, identify those that might be re-used, and publish via the architecture repository.
    - Document rationale for building block decisions in architecture document.
    - Prepare Applications Architecture report: Generate the Applications Architecture document. If appropriate, use reports and/or graphics generated by modeling tools to demonstrate key views of the architecture. Route the Applications Architecture document for review by relevant stakeholders, and incorporate feedback.
    - Checkpoint/Impact Analysis: Check the original motivation for the architecture project and the Statement of Architecture Work against the proposed Applications Architecture**.** Conduct an Impact Analysis, to:
        - Identify any areas where the Business Architecture (e.g., business practices) may need to change to cater for changes in the Applications Architecture (for example, changes to forms or procedures, application systems, or database systems).

          If the impact is significant, this may warrant the Business Architecture being revisited.
        - Identify any areas where the Data Architecture (if generated at this point) may need to change to cater for changes in the Applications Architecture (or to identify constraints on the Data Architecture, if about to be designed).

          If the impact is significant, this may warrant the Data Architecture being revisited, if already developed in this cycle.
        - Identify any constraints on the Technology Architecture about to be designed.
        - Refine the proposed Applications Architecture only if necessary.
8. **Gap analysis** and report:
    - Create gap matrix.
    - Identify building blocks to be carried over, classifying as either changed or unchanged.
    - Identify eliminated building blocks.
    - Identify new building blocks.
    - Identify gaps and classify as those that should be developed and those inherited.

Outputs

The outputs of this phase are:

- Statement of Architecture Work (updated if necessary)
- Applications Architecture Baseline Description, if appropriate

OMG: IT Portfolio Management Facility (ITPMF)

- Validated applications principles, or new applications principles (if generated here)

  Potentially represented by very general UML class diagrams.

- Target Applications Architecture:
  - Process systems model

    MDA standards: IT Portfolio Management Facility (ITPMF); UML (Activity Diagrams); Enterprise Distributed Object Computing (EDOC)
  - Place systems model

    MDA standard: IT Portfolio Management Facility (ITPMF)
  - Time systems model

    MDA standard: UML (Sequence Diagrams)
  - People systems model

    MDA standard: IT Portfolio Management Facility (ITPMF)
  - Applications interoperability requirements

    MDA standard: UML (Component Diagrams) or Enterprise Distributed Object Computing (EDOC)
- Viewpoints addressing key stakeholder concerns
- Views corresponding to the selected viewpoints; e.g.:
  - Common Applications Services view

    MDA standards: IT Portfolio Management Facility (ITPMF); UML (Component Diagrams) or Enterprise Distributed Object Computing (EDOC); Enterprise Application Integration (EAI)
  - Applications Interoperability view

    MDA standard: UML (Component Diagrams) or Enterprise Distributed Object Computing (EDOC); Enterprise Application Integration (EAI)
  - Applications/Information view

    MDA standards: IT Portfolio Management Facility (ITPMF); UML (Component Diagrams) or Enterprise Distributed Object Computing (EDOC)
  - Applications/User Locations view

    MDA standard: IT Portfolio Management Facility (ITPMF)
- Gap analysis results
- Applications Architecture report, summarizing what was done and the key findings
- Impact Analysis
  - Areas where the Business Architecture may need to change to cater for changes in the Applications Architecture
  - Identify any areas where the Data Architecture (if generated at this point) may need to change to cater for changes in the Applications Architecture
  - Constraints on the Technology Architecture about to be designed

    OMG: No mapping but views on other metamodels can be defined using MOF QVT.

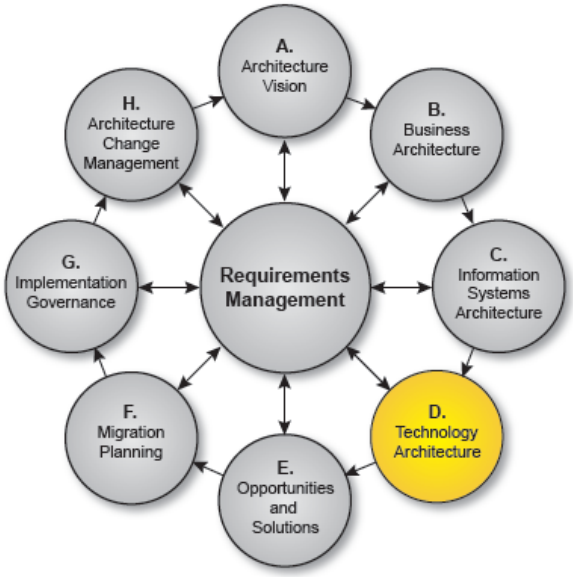
- Updated business requirements (if appropriate)

  MDA standard: SysML

Phase D: Technology Architecture

Objective

The objective of Phase D is to develop a Technology Architecture that will form the basis of the following implementation work.



Approach

Detailed guidelines for Phase D including Inputs, Steps, and Outputs, are given in Phase D: Technology Architecture – Detailed Description.

Architecture Continuum

As part of Phase D, the architecture team will need to consider what relevant Technology Architecture resources are available in the Architecture Continuum.

In particular:

- The TOGAF Technical Reference Model (TRM)
- Generic technology models relevant to your organization's industry "vertical" sector. For example:
  - The TeleManagement Forum (TMF) has developed detailed technology models relevant to the Telecommunications industry.
- Technology models relevant to common systems architectures. For example:
  - The Open Group has a Reference Model for Integrated Information Infrastructure (see Part III: Enterprise Continuum, III-RM) that focuses on the application-level components and underlying services necessary to provide an integrated information infrastructure.

Inputs

Inputs to Phase D are:

- Technical principles (if existing)
- Request for Architecture Work
- Statement of Architecture Work
- Architecture Vision
- Relevant technical requirements from previous phases
- Gap analysis (from Data Architecture)
- Gap analysis (from Applications Architecture)
- Baseline Business Architecture, Version 2 (detailed), if appropriate
- Data Architecture Baseline Description, if appropriate

- Applications Architecture Baseline Description, if appropriate
- Target Business Architecture, Version 2 (detailed)
- Re-usable building blocks (from organization's Enterprise Continuum, if available)
- Target Data Architecture
- Target Applications Architecture

Steps

1. **Technology Baseline Description:**
    - **Review Baseline Business Architecture**, **Baseline Data Architecture, and Baseline Applications Architecture**, to the degree necessary to inform decisions and subsequent work.
    - Develop a Baseline Description of the existing Technology Architecture, to the extent necessary to support the target Technology Architecture. The scope and level of detail to be defined will depend on the extent to which existing technology components are likely to be carried over into the Target Technology Architecture, and on whether existing architectural descriptions exist, as described in Approach. Define for each major hardware or software platform type:
        - Name (short and long)
        - Physical location
        - Owner(s)
        - Other users
        - Plain language description of what the hardware/software platform is and what it is used for
        - Business functions supported
        - Organizational units supported
        - Networks accessed
        - Applications and data supported
        - System inter-dependencies (for example, fall-back configurations)
    - To the extent possible, identify and document candidate Technology Architecture building blocks (potential re-usable assets).
    - Draft the Baseline Technology Architecture report: summarize key findings and conclusions, developing suitable graphics and schematics to illustrate baseline configuration(s). If warranted, provide individual Technology Architecture Baseline Descriptions as Annexes.
    - Route the Baseline Technology Architecture report for review by relevant stakeholders, and incorporate feedback. Refine the Baseline Description only if necessary.
2. **Target Technology Architecture:** See detailed steps. Detailed activities for this step, including Inputs, Activities, and Outputs, are given in Phase D: Technology Architecture – Detailed Description.

Outputs

The outputs of Phase D are:

- Statement of Architecture Work (updated if necessary)
- Technology Architecture Baseline Description, if appropriate

  MDA standards: IT Portfolio Management Facility (ITPMF) or UML (Deployment)

- Validated technology principles, or new technology principles (if generated here)
- Technology Architecture Report, summarizing what was done and the key findings
- Target Technology Architecture, Version 1
    - Technology Architecture – target services (a description of the service portfolios required, also known as an Organization Specific Framework)

      MDA standards: *IT Portfolio Management Facility (ITPMF), UML, or Enterprise Distributed Object Computing (EDOC)*
    - Technology Architecture – requirements traceability (business objectives criteria)

      MDA standards: *IT Portfolio Management Facility (ITPMF) or SysML*
    - Technology Architecture – architecture specification

      MDA standards: Appropriate metamodels
    - Technology Architecture – mapping of the architectures in the Architecture Continuum

      MDA standards: IT Portfolio Management Facility (ITPMF)
- Technology Architecture – gap report
- Viewpoints addressing key stakeholder concerns
- Views corresponding to the selected viewpoints:
    - Networked Computing/Hardware view

      MDA standards: *IT Portfolio Management Facility (ITPMF) or UML (Deployment)*
    - Communications view

      MDA standards: *IT Portfolio Management Facility (ITPMF) or UML (Deployment)*
    - Processing view

      MDA standards: *UML*

Phase D: Technology Architecture – Detailed Description

Introduction

This is the detailed description of the process to develop the Target Technology Architecture.

Overview

The steps involved in development of the Target Technology Architecture are illustrated in Figure 4.

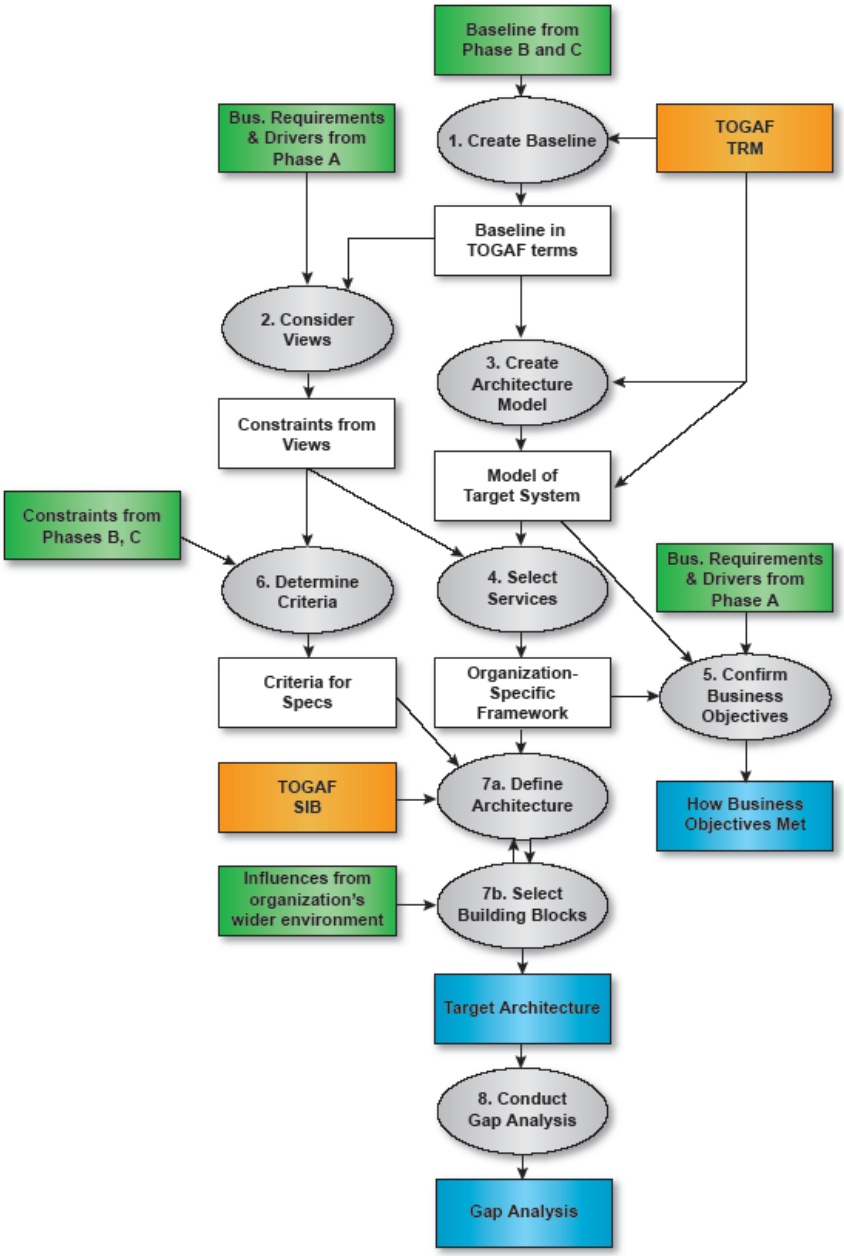


Figure 4: Technology Architecture Development

The boxes and ovals in Figure 4 are color-coded as follows:

- **Green** boxes represent inputs to the Technology Architecture development process from earlier phases in the architecture development cycle. This implies that some previous implementation exists for which an architecture may or may not have been defined previously.
- **Orange** boxes represent inputs to the process from TOGAF.
- White boxes represent inputs and outputs internal to the Technology Architecture development process.
- **Blue** boxes represent outputs to later phases in the architecture development cycle.
- **Gray** ovals represent phases within the Technology Architecture development process.

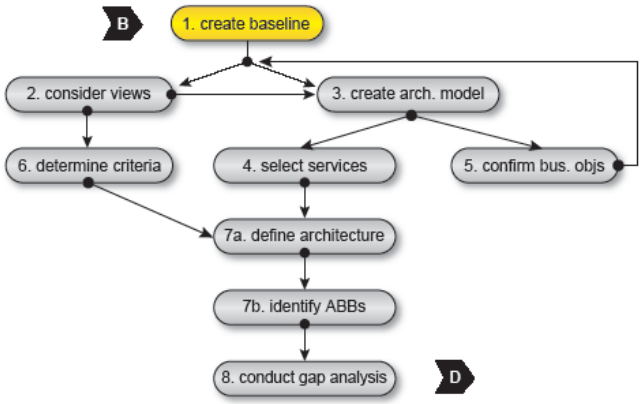
It is possible that an organization creating or adapting a Technology Architecture already mandates the use of a list of approved suppliers/products for that organization. Such a list would be an input to the definition of the organization-specific architecture framework, as shown in Figure 4. The architectures can then be used as procurement tools to govern future growth and the development of the organization's IT infrastructure.

The steps outlined in Figure 4 are expanded in the following subsections.

Phase D/Step 1: Create a Baseline Description in the TOGAF Format

Objective

The objective of this step is to convert the description of the existing system into services terminology using the organization's Foundation Architecture; e.g., the TOGAF Foundation Architecture's Technical Reference Model (TRM). The rationale behind this is to structure the existing system description in a way which makes it compatible with the breakdown of standards and the descriptions used within your Foundation Architecture.



Approach

This step is intended to facilitate moving from product documentation to a service-oriented description. The step will aid in specifying standards for the Target Architecture in Step 4. An additional step, Step 3, oriented to defining building blocks, provides the means to cross-check the architectural definition process in the form of implementation-related decisions.

Additionally, this step captures relevant parts of the existing architecture (using the scope definition established in Phase A) as candidates for re-usable building blocks, along with inhibitors to meeting business requirements using the existing system. The existing architecture is assessed against the Business Architecture, identifying the key inhibitors and opportunities for re-use. Finally, the existing architecture assessment ends with the capture of implied or explicit architecture principles that should be carried forward and imposed on this architecture exercise.

Begin by converting the description of the existing environment into the terms of your organization's Foundation Architecture (e.g., the TOGAF Foundation Architecture's TRM). This will allow the team developing the architecture to gain experience with the model and to understand its component parts. The team may be able to take advantage of a previous architectural definition, but it is assumed that some adaptation may be required to match the architectural definition techniques described as part of this process. Another important task is to set down a list of key questions which can be used later in the development process to measure the effectiveness of the new architecture.

A key process in the creation of a broad architectural model of the target system is the conceptualization of Architectural Building Blocks. Architectural Building Blocks are not intended to be solutions, but depictions of how the architecture might be looked on in implementable terms. Their functionality is clearly defined, but without the detail introduced by specific products. The method of defining Architectural Building Blocks, along with some general guidelines for their use in creating an architectural model, is described in Part IV: Resource Base, Building Blocks, Building Blocks and the ADM, and is illustrated in detail in the Building Blocks Example.

It is recommended that Architectural Building Blocks be documented (e.g., with an architecture description language) and stored (e.g., in a repository or information base), in order to maximize re-

use potential.

Applying the Architectural Building Block method introduces application space into the architectural process. This is the means of linking services, which address functionality that must be considered on an enterprise basis, with applications, which may or may not address global functionality. The Building Blocks Example in Part IV: Resource Base provides insight into both application-specific and more global considerations in defining building blocks in order to illustrate this.

Inputs

The inputs to Step 1 are:

- Technical principles (if existing)
- Request for Architecture Work
- Statement of Architecture Work
- Architecture Vision
- Relevant technical requirements from previous phases
- Gap analysis (from Data Architecture)
- Gap analysis (from Applications Architecture)
- Baseline Business Architecture, Version 2 (detailed), if appropriate
- Data Architecture Baseline Description, if appropriate
- Applications Architecture Baseline Description, if appropriate
- Business Architecture, Version 2 (detailed)
- Re-usable building blocks (from organization's Enterprise Continuum, if available)
- Target Data Architecture
- Target Applications Architecture
- Re-usable Architectural Building Blocks (from organization's Architecture Continuum, if available)
- Re-usable solutions building blocks (from organization's Solutions Continuum, if available)

Activities

Key activities in Step 1 include:

1. Collect data on current system.
2. Document all constraints.
3. Review and validate (or generate, if necessary) the set of Technology Architecture principles.
4. These will normally form part of an overarching set of architecture principles. Guidelines for developing and applying principles, and a sample set of Technology Architecture principles, are given in Part IV: Resource Base, Architecture Principles.
5. List distinct functionality.
6. Produce affinity groupings of functionality using TOGAF TRM service groupings (or your business' Foundation Architecture).
7. Analyze relationships between groupings.

8. Sanity check functionality to assure all of current system considered.
9. Identify interfaces.
10. Produce Technology Architecture model.
11. Verify Technology Architecture model.
12. Document key questions to test merits of Technology Architecture.
13. Document criteria for selection of service portfolio architecture.

Outputs

The outputs of Step 1 are:

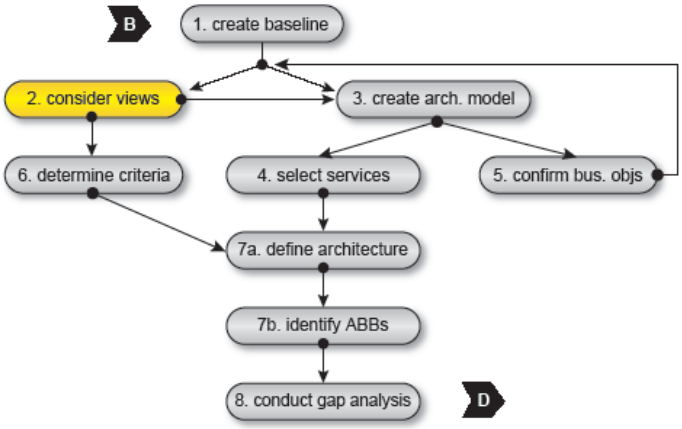
- Technical principles (if not existing)
- Technology Architecture, Version 0.1:
  - Technology Architecture – constraints
  - Technology Architecture – architecture principles
  - Technology Architecture – requirements traceability – key questions list
  - Technology Architecture – requirements traceability – criteria for selection of service portfolio
  - Technology Architecture Model, Version 0.1

    MDA: IT Portfolio Management Facility (ITPMF) or UML (Deployment)

Phase D/Step 2: Consider Different Architecture Reference Models, Viewpoints, and Tools

Objective

The objective of this step is to perform an analysis of the Technology Architecture from a number of different concerns (requirements) or viewpoints and to document each relevant viewpoint. The purpose of considering these viewpoints is to ensure that all **relevant** stakeholder concerns will have been considered in the final Technology Architecture, so ensuring that the target system will meet all the requirements put on it.



Approach

The Business Architecture is used to select the most relevant viewpoints for the project. It is important to recognize that in practice it will be rarely possible to reach 100% coverage of stakeholder concerns.

Pertinent viewpoints are created first from the existing system to identify the salient elements of the current systems requirements that the stakeholders confirm must also be satisfied in the target system. A comprehensive set of stakeholder viewpoints must also be created for the target system. The corresponding views of the existing system will be compared with the views of the target system to identify elements of the existing system that are intended for replacement or improvement.

If a set of viewpoints is carefully chosen, it will expose the most important aspects of the existing architecture and the requirements of the target system.

Several different viewpoints may be useful. Architecture viewpoints and views are described in greater detail in Part IV: Resource Base, Developing Architecture Views. The viewpoints presented there should not be considered an exhaustive set, but simply a starting point. In developing an Technology Architecture, it is very likely that some of the viewpoints given there will not be useful, while others not given there will be essential. Again, use the Business Architecture as a guide in selecting the pertinent viewpoints.

Inputs

The inputs to Step 2 are:

- Request for Architecture Work
- Statement of Architecture Work
- Business Architecture, Version 2
- Technology Architecture, Version 0.1

Activities

Key activities in Step 2 include:

1. **Select relevant Technology Architecture resources** (reference models, patterns, etc.) from the Architecture Continuum, on the basis of the business drivers, and the stakeholders and concerns.

   MDA standard: *Reusable Asset Specification (RAS) for re-use.*

2. **Select relevant Technology Architecture viewpoints** – i.e., those that will enable the architect to demonstrate how the stakeholder concerns are being addressed in the Technology Architecture. (See Part IV: Resource Base, Architecture Views, for examples).
   - Document the selected viewpoints, if not already documented.
     - Consider using ANSI/IEEE Std. 1471-2000 as a guide for documenting a viewpoint.
   - A primary reference model will be the TOGAF Technical Reference Model (TRM). Other reference models will be taken from the Architecture Continuum.
   - Consider developing at least the following views:
     - Networked Computing/Hardware view
     - Communications view
     - Processing view
     - Cost view
     - Standards view
   - Brainstorm and document technical constraints deriving from analysis of the concerns, and ensure they are covered by the viewpoints.
3. **Identify appropriate tools and techniques** to be used for capture, modeling, and analysis, in association with the selected viewpoints. Depending on the degree of sophistication warranted, these may comprise simple documents or spreadsheets, or more sophisticated modeling tools and techniques.

   MDA standards: Support for MDA can help guide what is "appropriate" for tools.

4. **Perform trade-off analysis** to resolve conflicts (if any) among the different viewpoints.

   One method of doing this is CMU/SEI's Architecture Trade-off Analysis (ATA) Method.

Outputs

The outputs of Step 2 are:

- Technology Architecture, Version 0.2
  - Technology Architecture – architecture viewpoints
    - Networked Computing/Hardware view

      MDA standards: *IT Portfolio Management Facility (ITPMF) or UML (Deployment)*
    - Communications view

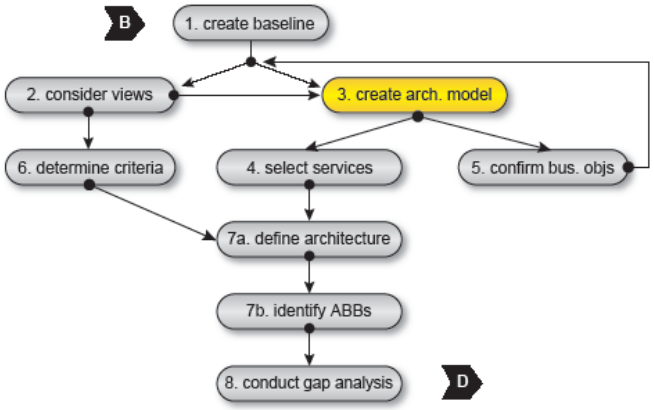
      MDA standards: *IT Portfolio Management Facility (ITPMF) or UML (Deployment)*
    - Processing view

      MDA standards: *UML*
    - Cost view

  - Standards view
- Technology Architecture – constraints

Phase D/Step 3: Create an Architectural Model of Building Blocks

Objective



The reason for selecting viewpoints in Step 2 is to be able to develop views for each of those viewpoints in Step 3. The architectural model created in Step 3 comprises those several views.

The objective of this step is to broadly determine how the services required in the target system will be grouped after considering all pertinent viewpoints of the architecture's use. This differs from Step 1 in that Step 1 dealt mainly with the required functionality of the system, whereas here we are considering many viewpoints that are not expressed explicitly as required functionality.

The rationale behind this is to enable the services required within the system to be selected during the next step, through the creation of an architecture model that clearly depicts the required services.

Approach

At Step 3, the purpose of examining different viewpoints in Step 2 becomes clear. The constraints defined and the unique system insights gained through an examination of the viewpoints pertinent to the current system and the target system can be used to validate the ability of the broad architectural model to accommodate the system requirements.

The broad architectural model starts as a TOGAF TRM-based model (or a model based upon the organization's Foundation Architecture), derived from the service-to-function mapping carried out as part of the service examination in Step 1. An architecture based exactly on the TOGAF TRM may not be able to accommodate the stakeholder needs of all organizations. If the examination of different viewpoints identifies architectural features that cannot be expressed in terms of the TOGAF TRM, changes and amendments to the TOGAF TRM should be made to create an organization-specific TRM.

Once the Baseline Description has been established and appropriate views described, it is possible to make decisions about how the various elements of system functionality should be implemented. This should only be in broad terms, to a level of detail which establishes how the major business functions will be implemented; for example, as a transaction processing application or using a client/server model.

Therefore this step defines the future model of building blocks (e.g., collections of functions and services generated from previous steps). It is here that re-use of building blocks from your business' Architecture Continuum is examined carefully, assuring that maximum re-use of existing material is realized.

Once the architecture model of building blocks is created, the model must be tested for coverage and completeness of the required technical functions and services. For each building block decision, completely follow through its impact and note the rationale for decisions, including the rationale for decisions not to do something.

Inputs

The inputs to Step 3 are:

- Business Architecture, Version 2
- Technology Architecture, Version 0.2
  - Technology Architecture – viewpoints
  - Technology Architecture – constraints
- Re-usable Architectural Building Blocks (from organization's Architecture Continuum, if available)

Activities

Key activities in Step 3 include:

1. To the extent possible, identify the relevant Technology Architecture building blocks, drawing on the Architecture Continuum.
2. For each viewpoint, create the model for the specific view required, using the selected tool or method. Consider developing at least the following views:
   - Networked Computing/Hardware view
   - Communications view
   - Processing view
   - Cost view
   - Standards view
3. Assure that all stakeholder concerns are covered. If they are not, create new models to address concerns not covered, or augment existing models.
4. Ensure that all information requirements in the Business Architecture, Data Architecture, and Applications Architecture are met.
5. Perform trade-off analysis to resolve conflicts (if any) among the different views.

   One method of doing this is CMU/SEI's Architecture Trade-off Analysis (ATA) Method.
6. Validate that the models support the principles, objectives, and constraints.
7. Note changes to the viewpoint represented in the selected models from the Architecture Continuum, and document.
8. Identify solution building blocks that would be used to implement the system, and create a model of building blocks.
9. Check building blocks against existing library of building blocks and re-use as appropriate.
10. Test architecture models for completeness against requirements.
11. Document rationale for building block decisions in architecture document.

Outputs

The outputs of Step 3 are:

- Technology Architecture, Version 0.3
  - Technology Architecture Model
    - Networked Computing/Hardware view

      MDA standards: *IT Portfolio Management Facility (ITPMF) or UML (Deployment)*
    - Communications view

      MDA standards: *IT Portfolio Management Facility (ITPMF) or UML (Deployment)*
    - Processing view

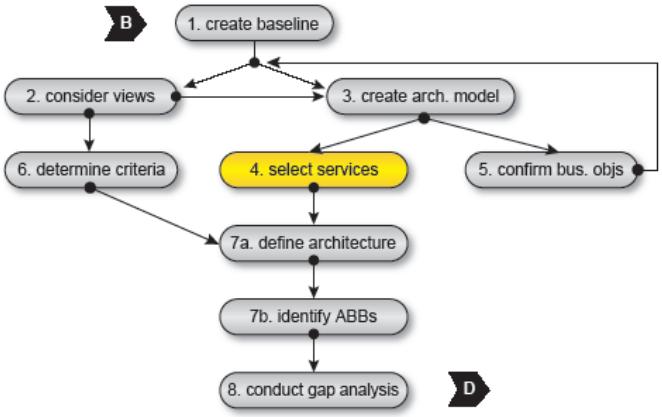
      MDA standards: *UML*
    - Cost view
    - Standards view
  - Technology Architecture – change requests and/or extensions or amendments to be incorporated in an organization-specific Architecture Continuum

    MDA standards: *Reusable Asset Specification (RAS) can be used to manage re-use aspects of building blocks.*

Phase D/Step 4: Select the Services Portfolio required per Building Block

Objective

The objective of this step is to select services portfolios for each building block generated in Step 3.



Approach

The services portfolios are combinations of basic services from the service categories in the TOGAF Technical Reference Model (TRM) that do not conflict. The combination of services are again tested to ensure support for the applications. This is a pre-requisite to the later step of defining the architecture fully.

The constraints output from Step 2 can provide more detailed information about:

- Requirements for organization-specific elements or pre-existing decisions (as applicable)
- Pre-existing and unchanging organizational elements (as applicable)
- Inherited external environment constraints

Where requirements demand definition of specialized services that are not identified in TOGAF, consideration should be given to how these might be replaced if standardized services become available in the future.

For each Architectural Building Block build up a service description portfolio as a set of non-conflicting services. The set of services must be tested to ensure that the functionality provided meets application requirements.

Inputs

The inputs to Step 4 are:

- Business Architecture, Version 2
- Technology Architecture, Version 0.3
- Technical Reference Model (TRM)
- Standards Information Base (SIB)

Activities

Key activities in Step 4 include:

1. Produce affinity grouping of services.
2. Cross-check affinity groups against needs.
3. Document service description portfolio for each Architectural Building Block cross-checking

for non-conflicting services.

4. Document change requests to architectures in the Architecture Continuum.

Outputs

The outputs of Step 4 are:

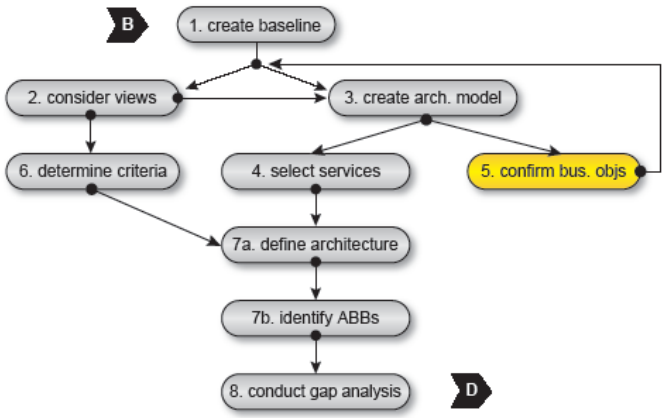
- Technology Architecture, Version 0.4
  - Technology Architecture – target services (a description of the service portfolios required also known as an Organization Specific Framework)

    MDA standards: IT Portfolio Management Facility (ITPMF), UML, or Enterprise Distributed Object Computing (EDOC)
  - Technology Architecture – change requests and/or extensions or amendments to be incorporated in an organization-specific Architecture Continuum

Phase D/Step 5: Confirm that the Business Goals and Objectives are Met

Objective

The objective of this step is to clarify and check the business goals and other objectives of implementing the architecture. This is required as a cross-check that the Technology Architecture meets these objectives.



Approach

The key question list is used to pose questions against the architecture model and service description portfolio to test its merit and completeness.

Inputs

The inputs to Step 5 are:

- Business Architecture (business goals), Version 2
- Technology Architecture, Version 0.4

Activities

Key activities in Step 5 include:

1. Conduct a formal checkpoint review of the architecture model and building blocks with stakeholders, validating that business goals are met. Utilizing the key questions list, ensure that the architecture addresses each question.
2. Document findings.

Outputs

The outputs of Step 5 are:

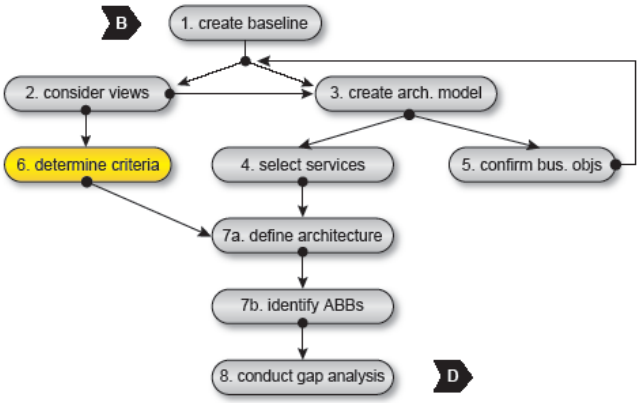
- Technology Architecture, Version 0.5
  - Technology Architecture – requirements traceability (business objectives criteria)

    MDA standards: IT Portfolio Management Facility (ITPMF) or SysML

Phase D/Step 6: Determine Criteria for Specification Selection

Objective

The objective of this step is to develop a set of criteria for choosing specifications and portfolios of specifications.



Approach

Choosing the right criteria is vital if the final architecture is to meet its objectives. These criteria will depend on the existing system and the overall objectives for the new architecture. The overall objectives should be developed from the organization's business goals, so it is hard to give specific advice here, but some example objectives are listed in Part IV: Resource Base, Business Scenarios.

Here are some example criteria, selected by a large government organization with the intention of building a stable and widely applicable architecture:

"A standard or specification:

- Must meet the organization's requirements
- Must meet legal requirements
- Should be a publicly available specification
- Should have been developed by a process which sought a high level of consensus from a wide variety of sources
- Should be supported by a range of readily available products
- Should be complete
- Should be well understood, mature technology
- Should be testable, so that components or products can be checked for conformance
- Should support internationalization
- Should have no serious implications for ongoing support of legacy systems
- Should be stable
- Should be in wide use
- Should have few, if any problems or limitations"

A high level of consensus is often considered the most important factor by large organizations because standards and specifications chosen have to accommodate a wide range of user needs. For example, in determining the level of consensus for standards in their architecture, the Application Portability Profile (APP), the US National Institute for Standards and Technology (NIST) prefers to use international standards for the basis of specifications. The process through which these international standards have evolved requires a very high level of consensus. A number of US Federal Information

Processing Standards (FIPS) specified in the APP are based on approved international standards. The use of international standards has significant benefits for any organization which works or trades with organizations in other countries.

Inputs

The inputs to Step 6 are:

- Business Architecture, Version 2
- Technology Architecture, Version 0.5
- Standards Information Base (SIB)

Activities

Key activities in Step 6 include:

1. Brainstorm criteria for choosing specifications and portfolios of specifications relying on previously used criteria for existing system and extrapolating for new architectural elements.
2. Meet with sponsors and present current state to negotiate a continue request from sponsors.

Outputs

The outputs of Step 6 are:

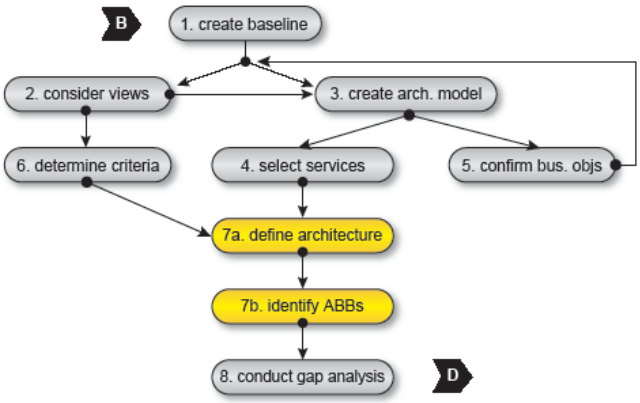
- Technology Architecture, Version 0.6
  - Technology Architecture – requirements traceability (standards selection criteria)

    MDA standards: IT Portfolio Management Facility (ITPMF) or SysML

Phase D/Step 7: Complete the Architecture Definition

Objective

The objective of this step is to fully specify the Technology Architecture. This is a complex and iterative process in which the selection of building blocks and interfaces has a big impact on how the original requirements are met. The figure shows this as two boxes, captioned as Steps 7a and 7b, but in reality the process is more complicated. See Part IV: Resource Base, Building Blocks, for further details.



Approach

Completion of the architecture definition may be achieved in two steps, by defining an intermediate Transitional Architecture in addition to the final Target Architecture, if complexity of migration requires it.

The specification of building blocks as a portfolio of services is an evolutionary process:

- The earliest building block definitions start as relatively abstract ones, defined by standards and services that map most easily to the architectural framework. These building blocks are most probably Architectural Building Blocks.
- At this stage a model and a portfolio of services have been established. The next step is to select the set of specifications that provide the services and that can be combined as required to create the building blocks.
- During this final step in the development of building blocks it must be verified that the organization-specific requirements will be met. The development process must include recognition of dependencies and boundaries for functions and should take account of what products are available in the marketplace. There are architectural and related solution-oriented building blocks.
- An example of how this might be expressed can be seen in Part IV: Resource Base, Building Blocks, Building Blocks Example. Building blocks can be defined at a number of levels matching the degree of integration that best defines the architecture of the system at any stage.
    - Fundamental functionality and attributes – semantic, unambiguous including security capability and manageability
    - Interfaces – chosen set, supplied (APIs, data formats, protocols, hardware interfaces, standards)
    - Dependent building blocks with required functionality and named used interfaces
    - Map to business/organizational entities and policies
- Finally the building blocks become more implementation-specific as Solution Building Blocks and their interfaces become the detailed architecture specification. Solution Building Blocks are

a means to determine how portions of the Target Architecture might be procured, developed, or re-used. The Solution Building Blocks architecture should have separate elements for developed, re-used, and procured building blocks, each described in terms of their minimum specification.

A full list of standards and specifications recommended by The Open Group can be found in Part III: Enterprise Continuum, Foundation Architecture – Standards Information Base.

Inputs

The inputs to Step 7 are:

- Business Architecture, Version 2
- Technology Architecture, Version 0.6
- Re-usable Architectural Building Blocks (from organization's Architecture Continuum, if available)
- Standards Information Base (SIB)

Activities

Key activities in Step 7 include:

1. Ensure clear documentation of all interfaces for each building block (APIs, data formats, protocols, hardware interfaces).
2. Select standards for each of the Architectural Building Blocks, re-using as much as possible from the reference models selected from the Architecture Continuum.
3. Fully document each Architectural Building Block.
4. Final cross-check of overall architecture against business requirements. Document rationale for building block decisions in architecture document.
5. Document final requirements traceability reports
6. Document final mapping of the architecture within the Architecture Continuum. From the selected Architectural Building Blocks, identify those that might be re-used, and publish via the architecture repository.
7. Document rationale for building block decisions in architecture document.
8. Generate the Technology Architecture document.
9. Prepare Technology Architecture report. If appropriate, use reports and/or graphics generated by modeling tools to demonstrate key views of the architecture. Route the Technology Architecture document for review by relevant stakeholders, and incorporate feedback.
10. Checkpoint/Impact Analysis: Check the original motivation for the architecture project and the Statement of Architecture Work against the proposed Technology Architecture**.** Conduct an Impact Analysis, to:
    - Identify any areas where the Business Architecture (e.g., business practices) may need to change to cater for changes in the Technology Architecture.

      If the impact is significant, this may warrant the Business Architecture being revisited.
    - Identify any areas where the Data Architecture may need to change to cater for changes in the Technology Architecture.

If the impact is significant, this may warrant the Data Architecture being revisited.

- Identify any areas where the Applications Architecture may need to change to cater for changes in the Technology Architecture.

  If the impact is significant, this may warrant the Applications Architecture being revisited.

- Refine the proposed Technology Architecture only if necessary.

Outputs

The outputs of Step 7 are:

- Technology Architecture, Version 0.7
  - Technology Architecture – architecture specification

    MDA standards: Appropriate metamodels
  - Technology Architecture – requirements traceability

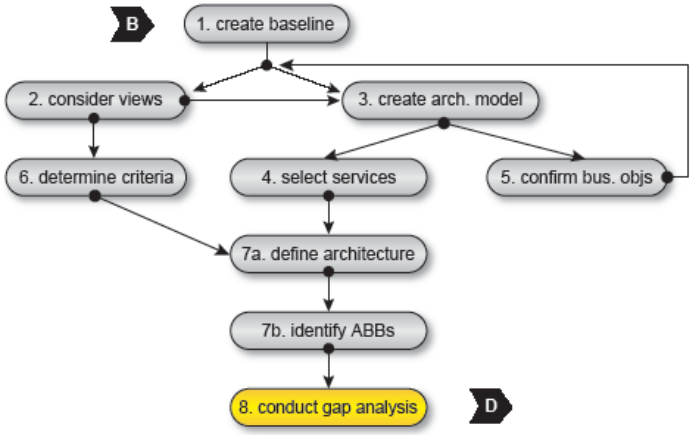
    MDA standards: IT Portfolio Management Facility (ITPMF) or SysML
  - Technology Architecture – mapping of the architectures in the Architecture Continuum

    MDA standards: IT Portfolio Management Facility (ITPMF)
- Technology Architecture report

Phase D/Step 8: Conduct a Gap Analysis

Objective

The objective of this step is to identify areas of the current and target system for which provision has not been made in the Technology Architecture. This is required in order to identify projects to be undertaken as part of the implementation of the target system.



Approach

A key step in validating an architecture is to consider what may have been forgotten. The architecture must support all of the essential information processing needs of the organization, as driven by the required applications. The most critical source of gaps that should be considered is stakeholder concerns that have not been addressed in subsequent architectural work.

Gap analysis highlights services and/or functions that have been accidentally left out, deliberately eliminated, or are yet to be developed or procured:

- Draw up a matrix with all the business functions of the current architecture on the vertical axis, and all the business functions of the Target Technology Architecture on the horizontal axis. In creating the matrix, it is imperative to use terminology that is accurate and consistent.
- Add to the Current Architecture axis a final row labeled “New Services”, and to the Target Architecture axis a final column labeled “Eliminated Services”.
- Where a function is available in both the current and Target Architectures, record this with “Included” at the intersecting cell.
- Where a function from the current architecture is missing in the Target Architecture (in the example, “broadcast services” and “shared screen services”), each must be reviewed. If it was correctly eliminated, mark it as such in the appropriate “Eliminated Services” cell. If it was not, you have uncovered an accidental omission in your new architecture that must be addressed by reinstating the function in the next iteration of the design – mark it as such in the appropriate “Eliminated Services” cell.
- Where a function from the Target Architecture cannot be found in the current architecture (in the example, “mailing list services”), mark it at the intersection with the “New” row, as a gap that needs to filled, either by developing or procuring the function.

When the exercise is complete, anything under “Eliminated Services” or “New Services” is a gap, which should either be explained as correctly eliminated, or marked as to be addressed by reinstating or developing/procuring the function.

Figure 5 shows an example from the Network Services category when functions from the current architecture are missing from the Target Architecture:

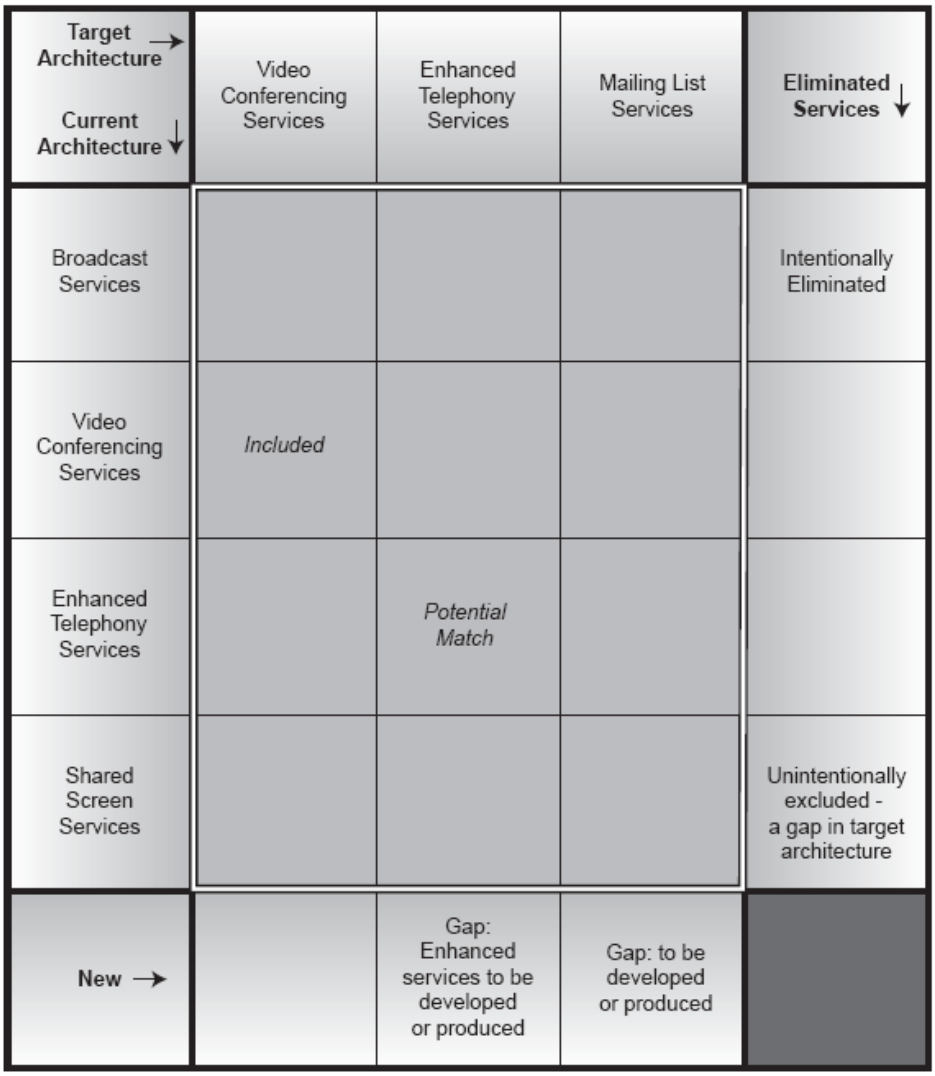


Figure 5: Gap Analysis Matrix

Inputs

The inputs to Step 8 are:

- Business Architecture, Version 2
- Data Architecture
- Applications Architecture
- Technology Architecture, Version 0.7

Activities

Key activities in Step 8 include:

1. Create gap matrix.
2. Identify building blocks to be carried over, classifying as either changed or unchanged.

3. Identify eliminated building blocks.
4. Identify new building blocks.
5. Identify gaps and classify as those that should be developed, those that should be procured, and those inherited.

Outputs

The output of Step 8 is:

- Technology Architecture, Version 1
  - Technology Architecture – gap report

Postscript

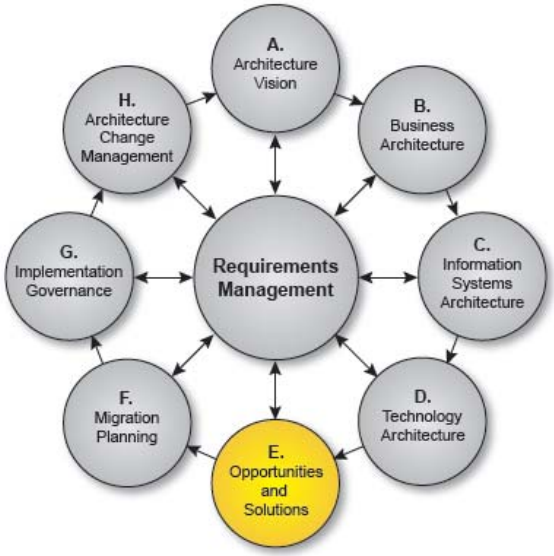
The Technology Architecture development process described above includes iterations. Financial and timing constraints should explicitly limit the number of iterations within steps 1 through 8, and drive to implementation. After that, a new cycle of architecture evolution may ensue.

Choosing the scope of an architecture development cycle carefully will accelerate the pay-back. In contrast, an excessively large scope is unlikely to lead to successful implementation.

"How do you eat an elephant? – One bite at a time."

Phase E: Opportunities and Solutions

Objective



The objectives of Phase E are to:

- Evaluate and select among the implementation options identified in the development of the various Target Architectures (for example, build *versus* buy *versus* re-use options, and sub-options within those major options)
- Identify the strategic parameters for change, and the top-level work packages or projects to be undertaken in moving from the current environment to the target
- Assess the dependencies, costs, and benefits of the various projects
- Generate an overall implementation and migration strategy and a detailed Implementation Plan

Approach

Phase E identifies the parameters of change, the major phases along the way, and the top-level projects to be undertaken in moving from the current environment to the target. The output of Phase E will form the basis of the Implementation Plan required to move to the Target Architecture. This phase also attempts to identify new business opportunities arising from the architecture work in previous phases.

Sometimes the process of identifying implementation opportunities allows a business to identify new applications, and in this case it may be necessary to iterate between Phase E and previous phases. Iteration must be limited by time or money to avoid wasting effort in the search for a perfect architecture.

Phase E is the first phase which is directly concerned with implementation. The task is to identify the major work packages or projects to be undertaken.

An effective way to do this is to use the gap analysis on the business functions between the old environment and the new, created in Phase D. Any functions appearing as "new" items will have to be implemented (developed or purchased and deployed).

Slightly harder to identify are the projects required to update or replace existing functions which must be done differently in the new environment. One of the options to be considered here is leaving an existing system in place and coexisting with the new environment.

During this final step in the specification of building blocks it must be verified that the organization-specific requirements will be met. Key to this is reason checking against the business scenario driving the scope of the project. It is important to note that the ensuing development process must include recognition of dependencies and boundaries for functions and should take account of what products

are available in the marketplace. An example of how this might be expressed can be seen in Part IV: Resource Base, Building Blocks, Building Blocks Example.

Coexistence appears on the surface to be easy. After all, the original system is left in place, largely unchanged. Unfortunately, it is not always as easy as it looks. The main problems with coexistence are:

- **User interfaces**: Combining user interfaces to the old and new applications in a single unit on the users' desks can be difficult, if not impossible.
- **Access to data**: Often the new applications need to share some data with the old applications, and some kind of data sharing must be established. This can be difficult unless the old and new systems use the same database technology.
- **Connectivity**: This may involve expenditure on software and gateway equipment. In difficult cases, equipment simply may not be available in a useful timescale. Often this happens because the old system is simply too out-of-date for connectivity solutions to be still on the market.

The most successful strategy for Phase E is to focus on projects that will deliver short-term pay-offs and so create an impetus for proceeding with longer-term projects.

Inputs

Inputs to Phase E are:

- Request for Architecture Work
- Statement of Architecture Work
- Business Architecture
- Data Architecture
- Applications Architecture
- Technology Architecture
- Re-usable Architectural Building Blocks (from organization's Enterprise Continuum, if available)
- Product information

Steps

Key steps in Phase E include:

1. Identify the key business drivers constraining the sequence of implementation (for example, reduction of costs, consolidation of services, introduction of new customer services, etc.).
2. Review the gap analysis generated in Phase D.
3. Brainstorm technical requirements from a functional perspective.
4. Brainstorm co-existence and interoperability requirements.
5. Perform architecture assessment and gap analysis.
6. Identify major work packages or projects, and classify as new development, purchase opportunity, or re-use of existing system.

Outputs

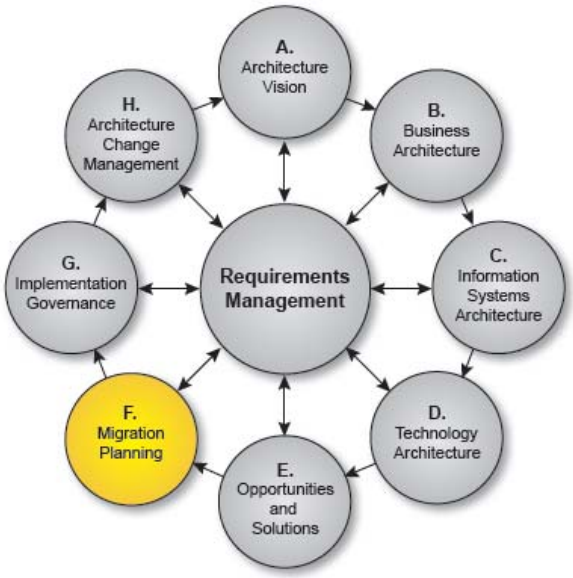
The outputs of Phase E are:

- Implementation and migration strategy
- High-level Implementation Plan
- Impact Analysis – project list

Phase F: Migration Planning

Objective

The objective of Phase F is to sort the various implementation projects into priority order. Activities include assessing the dependencies, costs and benefits of the various migration projects. The prioritized list of projects will go on to form the basis of the detailed Implementation Plan and Migration Plan.



Approach

There are some important questions to be asked before embarking on a migration exercise:

- What are the implications of this project on other projects and activities?
- What are the dependencies between this project and other projects and activities?
- What products are needed?
- What components must be developed?
- Does the organization have the resources needed to develop such components?
- What standards are the products or components built on?
- When will they be available?
- Will the products stand the test of time, both because of the technology they use and also because of the viability of the supplier?
- What is the cost of retraining the users?
- What is the likely cultural impact on the user community, and how can it be controlled?
- What is the total cost of the migration, and what benefits will it deliver? It is important to look at actual benefits, and not presumed benefits. Is the funding available?
- Is the migration viable?

Many things affect the answers to these questions, including the current and future architectures, the size of the organization and its complexity, and the value of technology to the core functions of the organization. Other things to consider are the asset value of the current systems, and the level of risk associated with changing the solution and/or the supplier.

Most organizations find that a change of architecture has too much impact on the organization to be undertaken in a single phase. Migration often requires consideration of a number of technical issues, not the least of which are those associated with the means of introducing change to operational systems.

Issues requiring special consideration may include:

- Parallel operations
- Choices of proceeding with phased migration by subsystem or by function
- The impact of geographical separation on migration

The decisions resulting from these considerations should be incorporated in the Implementation Plan.

There are a number of strategies for developing the Implementation Plan and Migration Plan.

The most successful basic strategy is to focus on projects that will deliver short-term pay-offs and so create an impetus for proceeding with longer-term projects.

One common approach is to implement business functions in a data-driven chronological sequence: i.e., create the applications and supporting technology that create data before those that process the data, before those that simply store, archive or delete data.

For example, the following detailed description of this approach is taken from SPE 68794, Implementing Enterprise Architecture – Putting Quality Information in the Hands of Oil and Gas Knowledge Workers.[12]

1. Determine the future disposition of current systems. Each current system is classified as:
    - **Mainstream systems** – part of the future information system.
    - **Contain systems** – expected to be replaced or modified in the planning horizon (next three years).
    - **Replace systems** – to be replaced in the planning horizon.

      The current system disposition decisions should be made by business people, not IT people.
2. Applications should be combined or split into parts to facilitate sequencing and implementation. This rearrangement of applications creates a number of projects, a project being equivalent to an application or to combinations or parts of applications.
3. Develop the data sequence for the projects as described in the Data Architecture. Using the CRUD (Create/Read/Update/Delete) matrix developed as part of the Data Architecture, sequence the projects such that projects that create data precede projects that read or update that data.
4. Develop an estimated value to the business for each project. To do this, first develop a matrix based on a value index dimension and a risk index dimension. The value index includes the following criteria: principles compliance, which includes financial contribution, strategic alignment, and competitive position. The risk index includes the following criteria: size and complexity, technology, organizational capacity, and impact of a failure. Each of the criteria has an individual weight. The index and its criteria and weighting are developed and approved by senior management early in the project. It is important to establish the decision-making criteria before the options are known.

In addition, there will be key business drivers to be addressed that will also tend to dictate the sequence of implementation, such as:

- Reduction of costs
- Consolidation of services

[12] G.A. Cox, R.M. Johnston, SPE, & R.M. Palermo, Aera Energy LLC, Copyright 2001, Society of Petroleum Engineers Inc.

- Ability to handle change
- A goal to have a minimum of "interim" solutions (they often become long-term/strategic!)

Another, possibly complementary, approach is for the individual projects or work packages to be group-sorted into a series of plateaux, each of which can be achieved in a realistic time scale.

The following description assumes a Target Architecture with only a single time horizon.

Inputs

Inputs to Phase F are:

- Request for Architecture Work
- Statement of Architecture Work
- Business Architecture, Version 2
- Technology Architecture
- Impact Analysis – project list

Steps

Key steps in Phase F include:

1. Prioritize projects
2. Estimate resource requirements and availability
3. Perform cost/benefit assessment of the various migration projects
4. Perform risk assessment
5. Generate implementation roadmap (time-lined)
6. Document the Migration Plan

Outputs

The output of Phase F is:

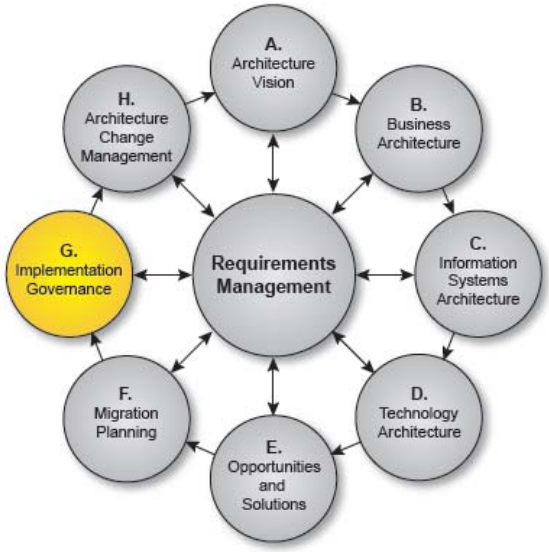
- Impact Analysis – detailed Implementation Plan and Migration Plan, including:
    - Architecture Implementation Contract (if appropriate)

Phase G: Implementation Governance

Objective

The objectives of Phase G are to:

- Formulate recommendations for each implementation project
- Construct an Architecture Contract to govern the overall implementation and deployment process
- Perform appropriate governance functions while the system is being implemented and deployed
- Ensure conformance with the defined architecture by implementation projects and other projects



Approach

It is here that all the information for successful management of the various implementation projects is brought together. Note that in parallel with Phase G there is the execution of an organizational-specific development process, where the actual development happens.

Phase G establishes the connection between architecture and implementation organization, through the Architecture Contract.

Project details are developed, including:

- Name, description, and objectives
- Scope, deliverables, and constraints
- Measures of effectiveness
- Acceptance criteria
- Risks and issues

Implementation governance is closely allied to overall Architecture Governance, which is discussed in Part IV: Resource Base, Architecture Governance.

A key aspect of Phase G is ensuring compliance with the defined architecture(s), not only by the implementation projects, but also by other ongoing projects within the enterprise. The considerations involved with this are explained in detail in Part IV: Resource Base, Architecture Compliance.

Inputs

Inputs to Phase G are:

- Request for Architecture Work
- Statement of Architecture Work

- Re-usable solutions building blocks (from organization's Solutions Continuum, if available)
- Impact Analysis – detailed Implementation Plan and Migration Plan (including Architecture Implementation Contract, if appropriate)

Steps

Key steps in Phase G include:

1. Project recommendation formulation – for each separate implementation project do the following:
    - Document scope of individual project in Impact Analysis
    - Document strategic requirements (from the architectural perspective) in Impact Analysis
    - Document change requests (such as support for a standard interface) in Impact Analysis
    - Document rules for conformance in Impact Analysis
    - Document time-line requirements from roadmap in Impact Analysis
2. Document Architecture Contract:
    - Obtain signature from all developing organizations and sponsoring organization
3. Ongoing implementation governance and Architecture Compliance review

Outputs

The output of Phase G is:

- Impact Analysis – implementation recommendations
- Architecture Contract, as recommended in Part IV: Resource Base, Architecture Contracts
- The architecture-compliant implemented system

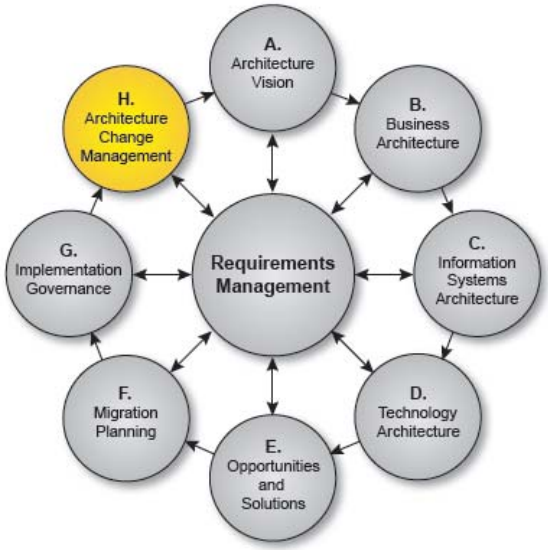
Note: The implemented system is actually an output of the development process. However, given the importance of this output, it is stated here as an output of the ADM. The direct involvement of architecture staff in implementation will vary according to organizational policy, as described in Part IV: Resource Base, Architecture Governance.

Phase H: Architecture Change Management

Objective

The objective of Phase H is to establish an architecture change management process for the new enterprise architecture baseline that is achieved with completion of Phase G. This process will typically provide for the continual monitoring of such things as new developments in technology and changes in the business environment, and for determining whether to formally initiate a new architecture evolution cycle.

Phase H also provides for changes to the framework and principles set up in the Preliminary Phase.



Approach

The goal of an architecture change management process is to ensure that changes to the architecture are managed in a cohesive and architected way, and to establish and support the implemented enterprise architecture as a *dynamic* architecture; that is, one having the flexibility to evolve rapidly in response to changes in the technology and business environment.

The change management process once established will determine:

- The circumstances under which the enterprise architecture, or parts of it, will be permitted to change after implementation, and the process by which that will happen
- The circumstances under which the enterprise architecture development cycle will be initiated again to develop a new architecture

The architecture change management process is very closely related to the architecture governance processes of the enterprise, and to the management of the Architecture Contract between the architecture function and the business users of the enterprise.

In Phase H it is critical that the governance body establish criteria to judge whether a change request warrants just an architecture update or whether it warrants starting a new cycle of the ADM. It is especially important to avoid "creeping elegance", and the governance body must continue to look for changes that relate directly to business value.

Guidelines for establishing these criteria are difficult to prescribe, as many companies accept risk differently, but as the ADM is exercised, the maturity level of the governance body will improve, and criteria will become clear for specific needs.

Drivers for Change

There are many technology-related drivers for architecture change requests. For example:

- New technology reports

- Asset management cost reductions
- Technology withdrawal
- Standards initiatives

This type of change request is normally manageable primarily through an enterprise's change management and architecture governance processes.

In addition there are business drivers for architecture change, including:

- Business-as-usual developments
- Business exceptions
- Business innovations
- Business technology innovations
- Strategic change

This type of change request often results in a complete re-development of the architecture, or at least in an iteration of a part of the architecture development cycle, as explained below.

The Change Management Process

The change management process needs to determine how changes are to be managed, what techniques are to be applied, and what methodologies used. The process also needs a filtering function that determines which phases of the architecture development process are impacted by requirements. For example, changes that affect only migration may be of no interest in the architecture development phases.

There are many valid approaches to change management, and various management techniques and methodologies that can be used to manage change; for example, project management methods such as PRINCE 2, service management methods such as ITIL, management consultancy methods such as Catalyst, and many others. An enterprise that already has a change management process in place in a field other than architecture (for example, in systems development or project management) may well be able to adapt it for use in relation to architecture.

The following describes an approach to change management, aimed particularly at the support of a dynamic enterprise architecture, which may be considered for use if no similar process currently exists.

The approach is based on classifying required architectural changes into one of three categories:

- **Simplification change**: A simplification change can normally be handled via change management techniques.
- **Incremental change**: An incremental change may be capable of being handled via change management techniques, or it may require partial re-architecting, depending on the nature of the change. See below for guidelines.
- **Re-architecting change**: A re-architecting change requires putting the whole architecture through the architecture development cycle again.

Another way of looking at these three choices is to say that a simplification change to an architecture is often driven by a requirement to reduce investment; an incremental change, by a requirement to derive additional value from existing investment; and a re-architecting change, by a requirement to

increase investment in order to create new value for exploitation.

To determine whether a change is simplification, incremental, or re-architecting, the following activities are undertaken:

1. Registration of all events that may impact the architecture
2. Resource allocation and management for architecture tasks
3. The process or role responsible for architecture resources assessment of what should be done
4. Evaluation of impacts

Guidelines for Maintenance versus Architecture Re-Design

A good rule-of-thumb is:

- If the change impacts two stakeholders or more, then it is likely to require an architecture re-design and re-entry to the ADM.
- If the change impacts only one stakeholder, then it is more likely to be a candidate for change management.
- If the change can be allowed under a dispensation, then it is more likely to be a candidate for change management.

For example:

- If the impact is significant for the business strategy, then there may be a need to redo the whole enterprise architecture – thus a re-architecting approach.
- If a new technology or standards emerge, then there may be a need to refresh the Technology Architecture, but not the whole enterprise architecture – thus an incremental change.
- If the change is at an infrastructure level – for example, ten systems reduced or changed to one system – this may not change the architecture above the physical layer, but it will change the baseline description of the Technology Architecture. This would be a simplification change handled via change management techniques.

In particular, a refreshment cycle (partial or complete re-architecting) may be required if:

- The Foundation Architecture needs to re-aligned with the business strategy.
- Substantial change is required to components and guidelines for use in deployment of the architecture.
- Significant standards used in the product architecture are changed which have significant end-user impact; e.g., regulatory changes.

If there is a need for a refreshment cycle, then a new Request for Architecture Work must be issued (to move to another cycle).

Inputs

Inputs to Phase H are:

- Request for Architecture Change – technology changes
  - New technology reports
  - Asset management cost reduction initiatives

  - Technology withdrawal reports
  - Standards initiatives
- Request for Architecture Change – business changes
  - Business developments
  - Business exceptions
  - Business innovations
  - Business technology innovations
  - Strategic change developments

Steps

Key steps in Phase H include:

- Ongoing monitoring of technology changes
- Ongoing monitoring of business changes
- Assessment of changes and development of position to act
- Meeting of Architecture Board (or other governing council) to decide on handling changes (technology and business)

Outputs

The outputs of Phase H are:

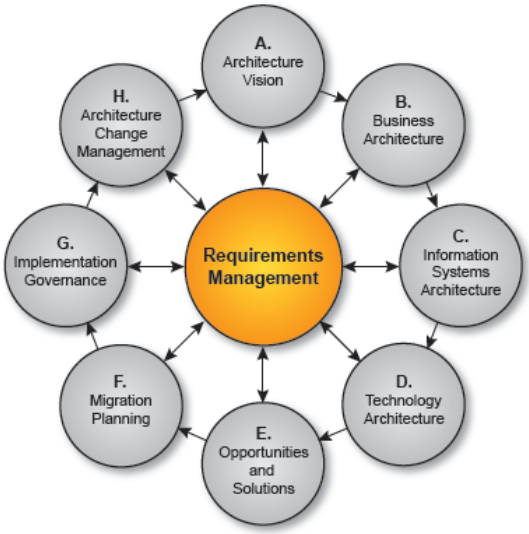
- Architecture updates

  MDA is a key enabler.
- Changes to architecture framework and principles
- New Request for Architecture Work (to move to another cycle)

ADM Architecture Requirements Management

Objectives

To define a process whereby requirements for enterprise architecture are identified, stored, and fed into and out of the relevant ADM phases.



Approach

As indicated by the "Requirements Management" circle at the center of the ADM graphic, the ADM is continuously driven by the requirements management process.

It is important to note that the "Requirements Management" circle denotes, not a static set of requirements, but a dynamic process whereby requirements for enterprise architecture and subsequent changes to those requirements are identified, stored, and fed into and out of the relevant ADM phases.

The ability to deal with changes in requirements is crucial. Architecture is an activity that by its very nature deals with uncertainty and change – the "grey area" between what stakeholders aspire to and what can be specified and engineered as a solution. Architecture requirements are therefore invariably subject to change in practice. Moreover, architecture often deals with drivers and constraints, many of which by their very nature are beyond the control of the enterprise (changing market conditions, new legislation, etc.), and which can produce changes in requirements in an unforeseen manner.

Note also that the requirements management process itself does not dispose of, address, or prioritize any requirements: this is done within the relevant phase of the ADM. It is merely the process for managing requirements throughout the overall ADM.

Resources

The world of requirements engineering is rich with emerging recommendations and processes for requirements management. TOGAF does not mandate or recommend any specific process or tool: it simply states what an effective requirements management process should achieve (i.e., the "requirements for requirements", if you like).

- Business Scenarios

  One effective technique that is described in TOGAF itself is business scenarios, which are an appropriate and useful technique to discover and document business requirements, and to articulate an architectural vision that responds to those requirements. Business scenarios are described in detail in Part IV: Resource Base, Business Scenarios.

- Volere Requirements Specification Template

  Architecture requirements is very much a niche area within the overall requirements field. One useful resource is the Volere Requirements Specification Template, available from the Volere web site hosted by the Atlantic Systems Guild. While not designed with architecture

requirements in mind, this is a very useful requirements template, which is freely available and may be modified or copied (for internal use, provided the copyright is appropriately acknowledged).

One interesting item in this template is the "waiting room", which is a hold-all for requirements in waiting. There are often requirements identified which, as a result of the prioritization activity that forms part of the requirements management process (see below), are designated as beyond the planned scope, or the time available, for the current iteration of the architecture. The waiting room is a repository of future requirements. Having the ability to store such requirements helps avoid the perception that they are simply being discarded, while at the same time helping to manage expectations about what will be delivered.

- Requirements Tools

  There is a large, and increasing, number of Commercial Off-The-Shelf (COTS) tools available for the support of requirements management, albeit not necessarily designed for architecture requirements. The Volere web site has a very useful list of leading requirements tools.

Inputs

The inputs to the requirements management process are the requirements-related outputs from each ADM phase.

The first high-level requirements are articulated as part of the Architecture Vision, generated by means of the business scenario or analogous technique.

Each architecture domain then generates detailed design requirements specific to that domain, and potentially to other domains (for example, areas where already designed architecture domains may need to change to cater for changes in this architecture domain; constraints on other architecture domains still to be designed.)

Deliverables in later ADM phases also contain mappings to the design requirements, and may also generate new types of requirements (for example, conformance requirements, time windows for implementation).

Steps

Key steps in the requirements management process include:

	Requirements Management Steps	ADM Phase Steps
1		Identify/document requirements – use business scenarios, or an analogous technique.
2	Baseline requirements: a. Determine priorities arising from current phase of ADM. b. Confirm stakeholder buy-in to resultant priorities. c. Record requirements priorities and place in requirements repository.	
3	Monitor baseline requirements	

4		Identify changed requirement: a. Remove or re-assess priorities. b. Add requirements and re-assess priorities. c. Modify existing requirements.
5	Identify changed requirement and record priorities: a. Identify changed requirements and ensure the requirements are prioritized by the architect(s) responsible for the current phase, and by the relevant stakeholders. b. Record new priorities. c. Ensure that any conflicts are identified and managed through the phases to a successful conclusion and prioritization. d. Generate Requirements Impact Statement for steering the architecture team. **Notes:** Changed requirements can come in through any route. To ensure that the requirements are properly assessed and prioritized, this process needs to direct the ADM phases and record the decisions related to the requirements. The requirements management phase needs to determine stakeholder satisfaction with the decisions. Where there is dissatisfaction, the phase remains accountable to ensure the resolution of the issues and determine next steps.	
6		a. Assess impact of changed requirements on current (active) phase. b. Assess impact of changed requirements on previous phases. c. Determine whether to implement change, or defer to later ADM cycle. If decision is to implement, assess timescale for change management implementation. d. Issue Requirements Impact Statement, Version n+1.
7		Implement requirements arising from Phase H. The architecture can be changed through its lifecycle by Phase H. The requirements management process ensures that new or changing requirements that are derived from Phase H are managed accordingly.
8	Update the requirements repository with information relating to the changes requested, including stakeholder views affected.	
9		Implement change in the current phase.

10		Assess and revise gap analysis for past phases. The gap analysis in the Phases B through D identifies the gaps between Baseline and Target Architectures. Certain types of gap can give rise to gap requirements. The ADM describes two kinds of gap: 1. Something that is present in the baseline, but not in the target (i.e., eliminated – by accident or design) 2. Something not in the baseline, but present in the target (i.e., new) A "gap requirement" is anything that has been eliminated by accident, and therefore requires a change to the Target Architecture. If the gap analysis generates gap requirements, then this step will ensure that they are addressed, documented, and recorded in the requirements repository, and that the Target Architecture is revised accordingly.

Outputs

The output of the requirements management process itself is:

- A **Structured Requirements Statement**, including:
  - Changed requirements
  - Requirements Impact Statement

The requirements repository contains the current requirements for the Target Architecture. When new requirements arise, or existing ones are changed, a Requirements Impact Statement is generated, which identifies the phases of the ADM that need to be revisited to address the changes. The statement goes through various iterations until the final version, which includes the full implications of the requirements (e.g., costs, timescales, business metrics) on the architecture development.

MDA standard: SysML

PART 3: MDA Standards

Introduction

Note: The information on MDA standards given in Part 3 is current at the time of writing, but is not maintained. The information is intended to provide context and detail to explain the types of MDA standard referenced in the ADM sections of this document, and the rationale for the specific mappings cited.

For current information on the status of any of the standards described here, refer to OMG's MDA web site.

MDA Specifications

Relevant MDA published and emerging specifications include the following:

Business Process Definition Metamodel (BPDM)

This forthcoming specification will provide a business process definition metamodel, which is platform-independent with respect to specific business process definition languages. The metamodel will define an abstract language for specification of executable business processes that execute within an enterprise (with or without human involvement), and may collaborate between otherwise-independent business processes executing in different business units or enterprises.

The specification will achieve the following:

- A common metamodel to unify the diverse business process definition graphical and textual notations that exist in the industry
- A metamodel that complements existing UML metamodels so that business processes specifications can be part of complete system specifications to assure consistency and completeness
- The ability to integrate process models for workflow management processes, automated business processes, and collaborations between business units
- Support for the specification of choreography, describing the collaboration between participating business entities using lightweight collaboration mechanisms (e.g., Web Services), and the ability to reconcile the choreography with supporting internal business processes
- The ability to exchange business process specifications between modeling tools, and between tools and execution environments using XMI

This specification will improve communication between modelers, including between business and software modelers, provide flexible selection of tools and execution environments, and promote the development of more specialized tools for the analysis and design of processes.

Business Semantics of Business Rules (BSBR)

There is no generally accepted approach for defining or representing business rules. The objective of this emerging specification (to be published as the *Semantics of Business Vocabulary and Rules* specification) is to allow business people to define the policies and rules by which they run their

business in their own language, in terms of the things they deal with in the business, and to capture those rules in a way that is clear, unambiguous, and readily translatable into other representations. Among those representations are presentation forms for business people and software engineers, and executable rules for many kinds of automated systems.

Business rules are used to make business decisions and provide control for business processes in a number of ways. As a consequence, there is a close relationship between business processes and business rules, and there are links between information systems that support rules and information systems that support business processes. Addressing this linkage, including rules as origins of events and rules as constraints on process transitions, is a requirement of the Business Process Definition Metamodel (BPDM). This proposed specification is expected to define the Business Rules Metamodel (BRM) – a metamodel for capturing the semantics of business rules.

Common Warehouse Metamodel (CWM) (CWM)

The primary objective of the CWM specification is to enable easy interchange of warehouse and business intelligence metadata between warehouse tools, warehouse platforms, and warehouse metadata repositories in distributed heterogeneous environments. CWM is based on three key industry standards: UML, MOF, and XMI. These three standards form the core of the OMG metadata architecture. Key aspects of the architecture include:

- A metamodeling architecture for general-purpose manipulation of metadata in distributed object repositories
- The use of UML notation for representing metamodels and models
- The use of standard information models (UML) to describe the semantics of analysis and design metamodels
- The use of MOF to define and manipulate metamodels programmatically grained CORBA interfaces; this approach leverages the strength of distributed object infrastructure
- The use of XMI for stream-based interchange of metadata

IT Portfolio Management Facility (ITPMF)

This specification addresses the need to manage deployable software and platform elements in the wider context of the business as a whole.

The fundamental intent of this specification is to supply only a core model that can be linked to any external model (or not at all). A Linkage package is specified for supporting this linking. In the future a user of a MOF 2.0 version of ITPMF could use model merging to incorporate the ITPMF classes directly into their metamodels.

The ITPMF is designed to be extensible and so Kind objects (which are akin to UML Stereotypes) are used extensively. To provide commonality of support for frequently used elements, a library of Kind instance elements is also specified.

It is a design aim that the specification should be implementable using conventional database approaches – so the design decision was taken not to assume the presence of UML.

Overall there are a number of options for extensibility:

- Extend the supplied model directly, by defining subclasses.

- Link with other models using the Linkage package or just defining new MOF Associations.
- Use the generic Kind and Property mechanisms within the specification itself; this means that this ITPMF specification can be used to create a stand-alone facility that is still extensible.

Meta Object Facility (MOF)

The Meta-Object Facility (MOF) specification defines an abstract language and a framework for specifying, constructing, and managing technology-neutral metamodels. A metamodel is in effect an abstract language for some kind of metadata. Examples include the metamodels for UML, CWM, and the MOF itself, as well as those in various OMG specifications in progress.

In addition, the MOF defines a framework for implementing repositories that hold metadata (e.g., models) described by the metamodels. This framework uses standard technology mappings to transform MOF metamodels into metadata APIs. This gives consistent and interoperable metadata repository APIs for different vendor product and different implementation technologies.

The MOF specification includes the following:

- A formal definition of the MOF meta-metamodel; that is, the abstract language for specifying MOF metamodels
- A set of "reflective" interfaces for managing metadata independent of the metamodel
- A set of interfaces for representing and managing MOF metamodels

UML and MOF are normally viewed in the context of a conceptual layered metadata architecture. Further to this, the metamodels for MOF and UML are designed to be architecturally aligned, sharing a common subset of core object modeling constructs. This alignment allows the MOF to re-use the UML notation for visualizing metamodels.

MOF 2.0 Query/Views/Transformations (QVT)

This specification addresses the need for standardizing the way mappings are achieved between models whose languages are defined using MOF. In addition, it specifies a standard way of querying MOF models, and creating views onto these models. This is similar to the need for XSLT for XML, where an XSLT script defines the mapping between a pair of DTDs and dictates how XMLs (compliant with the DTDs) are transformed.

Queries on MOF models are required both to filter and select elements from a model on an *ad hoc* basis, as well as to select elements that are the sources for transformations. This is similar to the need for Xpath within XSLT.

A view reveals specific aspects of a modeled system. A view is a model that is derived from another model. This specification provides a mechanism for creating views. It also addresses a common problem in current OMG specifications and in many emerging Java Community Process JSRs (such as JMI). In these specifications transformation rules are described in English text, BNF, and other mechanisms and there is no single standard way of defining them formally.

Ontology Definition Metamodel (ODM)

This emerging specification seeks to define a MOF 2.0 Metamodel, UML2 profile, and any additional information needed to support:

- Development of ontologies using UML modeling tools

- Implementation of ontologies in the W3C Web Ontology Language (OWL)
- Forward and reverse engineering for ontologies

Organization Structure Metamodel (OSM)

The objective of this emerging specification is to provide a metamodel to be used for specification of an organization structure and at least one mapping to a production directory schema. An organization structure metamodel consists of modeling elements used to represent organizational entities, their attributes, the relationships between them, and the people assigned to them. This includes information about the organizational units that make up an enterprise and the structure and formal relationships by which they interact.

This proposed metamodel will support the specification, analysis, and modification of the organization structure of an enterprise. The concepts and associated information must be defined in business terms and must be sufficient to support a variety of organization types. The metamodel will also support specification of future-effective organizational changes.

The specification will include a mapping of the metamodel to a run-time directory schema to provide consistent data structures for use by production applications while addressing the schema characteristics needed for run-time performance.

Reusable Asset Specification (RAS)

The scope of this specification is a set of guidelines and recommendations about the structure, content, and descriptions of re-usable software assets. Recognizing that there are different categories of re-usable software assets, the specification identifies some categories – or rather types or profiles – and provides general guidelines on these profiles.

RAS addresses the engineering elements of re-use. It attempts to reduce the friction associated with re-use transactions through consistent, standard packaging. This is much like the steering wheel, turn signals, pedals, and fuel gauge in a car; although they are slightly different across car models and makes, there is a familiarity among them that significantly reduces the costs of re-use.

Software Process Engineering Metamodel (SPEM)

This specification defines a metamodel used to describe a concrete software development process or a family of related software development processes. Process enactment is outside the scope of SPEM, although some examples of enactment are included for explanatory purposes.

The SPEM is a metamodel for defining systems engineering development processes and their components. A tool based on SPEM would be a tool for process authoring and customizing. The actual enactment of processes – that is, planning and executing a project using a process described with SPEM – will be addressed in the next release.

This specification is limited to defining the minimal set of process modeling elements necessary to describe any development process, without adding specific models or constraints for any specific area or discipline, such as project management or analysis.

The authors believe this is the appropriate approach for the software process engineering domain, and any attempt to standardize a more complex and detailed model at this time would be both unwise and ineffective. The standard wants to accommodate a large range of existing and described software

development processes, and not exclude them by having too many features or constraints.

Unified Modeling Language (UML)

A specification defining a graphical language for visualizing, specifying, constructing, and documenting the artifacts of distributed object systems.

UML Profile for Enterprise Application Integration (EAI)

As enterprises adapt to business change and new opportunities, they seek to build on their existing strengths and assets for competitive advantage. Electronic trading with consumers and other businesses is one of these trends. This frequently entails building new applications by coupling existing ones, which is known as Enterprise Application Integration (EAI). This is most often done with some form of messaging that provides loose coupling to make it easy to change, to link heterogeneous systems and operating environments, and to maximize resilience and robustness in cases of partial failure.

EAI technology is being promoted to integrate legacy systems with new packages. But integrating legacy applications with new software is a difficult and expensive task due, in large part, to the necessity of customizing each connection that ties together two disparate applications. There is no single mechanism to describe how one application may allow itself to be invoked by another.

This specification solves this problem by defining and publishing a metadata interchange standard for information about accessing application interfaces. The goal is to simplify application integration by standardizing application metadata for invoking and translating application information. Once these standards exist, tools may be constructed to facilitate the development, execution, and management of these integration points.

Such connected systems are inherently complex to define and manage. A well-known approach to managing complexity is to define levels of concern. Modeling with UML has been shown to be successful at representing differing levels of detail. The appropriate level for EAI is Applications Architecture – the treatment of the interfaces and interactions between applications. UML has been used successfully for modeling at this level, and this specification presents the authors' view of best practice for using the existing UML for modeling Applications Architectures; i.e., architectures composed by enterprises to enable application integration.

UML Profile for Enterprise Distributed Object Computing (EDOC)

The vision of the Enterprise Collaboration Architecture (ECA, an element of this specification) is to simplify the development of component-based EDOC systems by means of a modeling framework and conforming to the MDA that provide:

- A platform-independent, recursive collaboration-based modeling approach that can be used at different levels of granularity and different degrees of coupling, for both business and systems modeling and encompasses:
- A loosely coupled, re-usable business collaboration architecture that can be leveraged by business-to-business (b2b) and business-to-customer (b2c) applications, as well as for Enterprise Application Integration (EAI)
- A business component architecture that provides interoperable business components and services, re-use and composability of components, and re-use of designs and patterns, while being independent of choice of technology (e.g., component models), independent of choice of

middleware (e.g., message services), and independent of choice of paradigms (e.g., synchronous or asynchronous interactions)

- Modeling concepts for describing clearly the business processes and associated rules that the systems support, the application structure and use of infrastructure services, and the breakdown of the system into configurable components
- An architectural approach that allows the integration of "process models" and "information models"
- A development approach that allows two-way traceability between the specification, implementation, and operation of enterprise computing systems and the business functions that they are designed to support
- Support for system evolution and the specification of collaboration between systems
- A notation that is accessible and coherent

The vision addresses key business needs by enabling the development of tools that support:

- Business collaborations as a central concern – covering alliances, outsourcing, supply chains, and Internet commerce, and dealing with relationships that are in constant flux where what is inside the enterprise today is outside tomorrow, and *vice versa*.
- Process engineering by assembling services – so that basic business functions can remain relatively constant while who performs them and in what sequence changes, and services themselves can become proactive
- The ability for parts of the enterprise to react quickly and reliably to change through:
- Shorter development time and improved quality of applications meeting market needs, improved interoperability between systems, and support for distributed computing
- Reduced lead-time and improved quality resulting from the ability to generate a substantial portion of application code
- More robust specification by removing ambiguity and enabling more rigorous analysis of designs
- A new marketplace for interoperable collaboration-based infrastructures and business components

XML Metadata Interchange (XMI)

XMI is a metamodel-driven XML integration framework for defining, interchanging, manipulating, and integrating XML data and objects. XMI-based standards are in use for integrating tools, repositories, applications, and data warehouses. XMI provides rules by which a schema can be generated for any valid XMI-transmissible MOF-based metamodel.

XMI provides a mapping from MOF to XML. As MOF and XML technology evolved, the XMI mapping is being updated to comply with the latest versions of these specifications. Updates to the XMI mapping have tracked these version changes in a manner consistent with the existing XMI Production of XML Schema specification (XMI Version 2).

Other OMG Specifications of Interest

Business Motivation Metamodel (BMM)

The Business Motivation Model is a "lightweight" *de facto* standard that provides a metamodel for enterprise-specific models. An enterprise's Business Motivation Model:

- Contains and organizes the elements of its business plans – vision and mission, influences and assessments, goals and objectives, strategies and tactics, business policies
- References other relevant elements of its business models – its business processes, business rules, and organization units – that are contained in related models built using standards outside the scope of the Business Motivation Model

Business Process Runtime Interfaces Platform Independent Model (BPRI)

This emerging specification seeks to define a Platform-Independent Model (PIM) of the run-time interfaces to business processes. Business processes include workflow management, but address a broader range of applications, including processes supporting electronic commerce and Web Services. It will not include models to specify business processes, but only interfaces to business processes when they are executed.

A business process in this context is a set of activities coordinated and controlled by the execution of a business process definition. People, computer applications, or other business processes that together accomplish a business objective may perform these activities. Progress, and thus the current status of a business process, is established by committed states reflecting the initiation and completion of activities and occurrence of other key events in the execution of the process. These specifications for BPRI will define the platform-independent interfaces by which business processes are initiated, monitored, and controlled, and through which the processes interact to perform their activities. Furthermore, these specifications will specify mappings from the BPRI PIM to one or more existing Platform-Specific Models (PSM) or technologies in which business processes might be implemented.

Production Rule Representation (PRR)

This emerging specification addresses the representation of production rules in UML models. (Production rules should not be confused with XMI production rules as defined in the XMI specification or other model or grammar transformation rules specified by the OMG standards.) With respect to production rules, this specification will support:

- A MOF 2.0-compliant metamodel with precise dynamic semantics to represent production rules, where "production rules" refers to rules that are executed by an inference engine. This metamodel is intended to support a language that can be used with UML models for explicitly representing production rules as visible, separate, and primary model elements in UML models.
- An XMI W3C XML Schema Description (xsd) for production rules, based on the proposed metamodel, in order to support the exchange of production rules between modeling tools and inference engines.
- An example of a syntax that is compliant with the proposed metamodel for expressing production rules in UML models. This syntax will be considered non-normative.

In addition, OMG has a number of vertical Domain Task Forces developing models relevant to specific domains such as Healthcare, Transportation, Manufacturing, Finance, etc. Please consult the OMG web site for more information.

UML versus Specific Metamodels

In various points of the ADM, UML is cited as an alternative to the use of specific metamodels, The following are the considerations that apply in this choice:

- UML Profiles allow use of existing UML tools.
  - Note that Profiles change dramatically at UML2.
- However, the resultant elements are UML elements not EA elements (e.g., a Class with the Stereotype <<Org Unit>> attached not a Org Unit).
  - Hard to manage, query, create views, etc.
  - Hard to interchange
- Approaches (both could be automated!):
  - Create mappings/transformations between profiles and metamodels
  - Create metamodel-specific tools
- In either case the metamodel should be the start point.

Mapping of Specific MDA Standards to TOGAF: Summary

Business Process Definition Metamodel (BPDM)

- Architecture Vision/Baseline Business Architecture
- Architecture Vision/Business Architecture
- Business Architecture/Business Functions
- Business Architecture/Business Services
- Business Architecture/Business Processes
- Target Business Architecture, Version 2 (detailed)/Business Functions
- Target Business Architecture, Version 2/Business Services
- Target Business Architecture, Version 2/Business Processes, including measures and deliverables
- Target Business Architecture, Version 2/Correlation of Organization and Functions
- Data Architecture/Target Data Architecture/Data Entity – Business Function Matrix
  - (Generic) references from BPDM to CWM

Business Semantics of Business Rules(BSBR)

- Architecture Vision/Business Architecture
- Data Architecture/Data Principles

Organization Structure Metamodel (OSM)

- Architecture Vision/ Baseline Business Architecture
- Architecture Vision/Business Architecture
- Target Business Architecture, Version 2 (detailed)/Organization Structure, identifying business locations and relating them to organizational units
- Target Business Architecture, Version 2/Business Roles, including development and

modification of skills requirements

- Target Business Architecture, Version 2/Correlation of Organization and Functions (referenced from BPDM)

Business Motivation Metamodel (BMM)

- Architecture Vision/Statement of Architecture Work/Scope and Constraints
- Architecture Vision/Refined statements of Business Principles, Business Goals, and Strategic Drivers
- Architecture Vision/Architecture Principles (BMM policy elements)
- Architecture Vision/Business Architecture
- Business Architecture/Validated Business Principles, Business Goals, and Strategic Drivers
- Target Business Architecture, Version 2 (detailed)/Business Goals and Objectives, for each organizational unit
- Data Architecture/Data Principles (policy elements)

Enterprise Distributed Object Computing (EDOC)

- Architecture Vision/Technology Architecture
- Data Architecture/Target Data Architecture/Data Entity – Business Function Matrix
- Data Architecture/Target Data Architecture/Data Interoperability Requirements
- Applications Architecture/Target Applications Architecture/Process Systems Model
- Applications Architecture/Target Applications Architecture/Applications Interoperability Requirements
- Applications Architecture/Common Applications Services view
- Applications Architecture/Applications Interoperability view
- Applications Architecture/Applications/Information view
- Technology Architecture/Target Services

Enterprise Application Integration (EAI)

- Architecture Vision/Technology Architecture

Common Warehouse Metamodel (CWM) (CWM)

- Data Architecture/Data Architecture Baseline Description
- Data Architecture/Data Principles
- Data Architecture/Target Data Architecture/Conceptual Data Model
- Data Architecture/Target Data Architecture/Logical Data Model
- Data Architecture/Target Data Architecture/Data Management Process Models
- Data Architecture/Target Data Architecture/Data Interoperability Requirements
- Data Architecture/Data Dissemination view

Meta Object Facility (MOF)

- Data Architecture/Data Model Management view

Ontology Definition Metamodel (ODM)

- Data Architecture/Data Principles
- Data Architecture/Target Data Architecture/Conceptual Data Model
- Data Architecture/Target Data Architecture/Logical Data Model

Software Process Engineering Metamodel (SPEM)

- Preliminary Phase: Framework and Principles
  - Objective: To define the framework and detailed methodologies that are going to be used to develop enterprise architectures in the organization concerned (typically, an adaptation of the generic ADM). SPEM can be used to model the TOGAF process itself [SPEM2 at pre-submission stage].
  - Deliverable: Framework Definition
- Architecture Vision
  - Objective: To define the scope of, and to identify and prioritize the components of, the current architecture effort. Modeling the ADM process (the generic process, and/or the adapted, organization-specific one), by means of SPEM, would help the scoping definition process.
  - Objective: To understand the impact on, and of, other enterprise architecture development cycles ongoing in parallel. Holding these other enterprise architecture development cycles as SPEM models would greatly help this impact assessment.

Re-Usable Asset Specification (RAS)

- Preliminary Phase: Framework and Principles
  - Approach: The enterprise's approach to re-use of architecture assets is a key part of both the framework definition and architecture principles. (Typically the principles will state the policy on re-use; and the framework will explain how re-use is effected.) The RAS specification could be used to model re-use aspects of all the architectural assets held, and their inter-relationships, and provide access to the specific models and artifacts.
- Architecture Vision/Enterprise Continuum
  - Existing architectural documentation (framework description, architectural descriptions, existing baseline descriptions, etc.) could take the form of existing MDA models. The RAS specification could be used to model all the architectural assets held, and their inter-relationships; with links to the specific models.
- Technology Architecture/Change Requests
  - RAS can be used to manage re-use aspects of building blocks.

UML

- Architecture Vision/Technology Architecture
- Business Architecture/Baseline Business Architecture, Version 2 (detailed) (use-cases)
- Business Architecture/Views corresponding to the selected viewpoints addressing key stakeholder concerns
- Business Architecture/Technical Requirements (drivers for Technology Architecture work)

- Data Architecture/Target Data Architecture/Logical Data Model
- Applications Architecture/Target Applications Architecture/Process Systems Model
  - Activity Diagrams
- Applications Architecture/Target Applications Architecture/Systems Place Model
  - Deployment Diagrams
- Applications Architecture/Target Applications Architecture/Systems Time Model
  - Sequence Diagrams
- Applications Architecture/Target Applications Architecture/Applications Interoperability Requirements
  - Component Diagrams
- Applications Architecture/Common Applications Services view
  - Component Diagrams
- Applications Architecture/Applications Interoperability view
  - Component Diagrams
- Applications Architecture/Applications/Information view
  - Component Diagrams
- Technology Architecture/Architecture Model
  - Deployment Diagrams
- Technology Architecture/Networked Computing/Hardware view
  - Deployment Diagrams
- Technology Architecture/Communications view
  - Deployment Diagrams
- Technology Architecture/Processing view
- Technology Architecture/Target Services

SysML

- Architecture Vision/Business Requirements
- Business Architecture/Technical Requirements (drivers for Technology Architecture work)
- Business Architecture/Updated Business Requirements
- Data Architecture/Relevant Technical Requirements
- Data Architecture/Updated Business Requirements
- Applications Architecture/Updated Business Requirements
- Technology Architecture/Requirements Traceability
- Technology Architecture/Requirements Traceability (criteria)

IT Portfolio Management Facility (ITPMF) Metamodel

- Architecture Vision/Baseline Technology Architecture
- Architecture Vision/Technology Architecture
- Target Business Architecture, Version 2 (detailed)/Business Functions

- Target Business Architecture, Version 2/Correlation of Organization and Functions
- Data Architecture/Target Data Architecture/Data Entity – Business Function Matrix
- Data Architecture/Data Dissemination view
- Data Architecture/Data Security view
- Data Architecture/Relevant Technical Requirements
- Applications Architecture/ Baseline Applications Architecture
- Applications Architecture/Target Applications Architecture/Process Systems Model
- Applications Architecture/Target Applications Architecture/Systems Place Model
- Applications Architecture/Target Applications Architecture/People Systems Model
- Applications Architecture/Common Applications Services view
- Applications Architecture/Applications/Information view
- Applications Architecture/Applications – User Locations view
- Technology Architecture/Architecture Model
- Technology Architecture/Networked Computing/Hardware view
- Technology Architecture/Communications view
- Technology Architecture/Target Services
- Technology Architecture/Requirements Traceability
- Technology Architecture/Requirements Traceability (criteria)
- Technology Architecture/Mapping in Enterprise Continuum

QVT

- Data Architecture/Impact Analysis
  - Views on other metamodels
- Applications Architecture/Impact Analysis
  - Views on other metamodels

Proposed Next Steps

1. Decide about interoperability; e.g., MOF 2.0 *versus* MOF 1.0 basis.
2. Consider tooling issues:
   - Especially for metamodels with no UML Profile
3. Consider versioning issues:
   - Do incremental versions of TOGAF architecture map to versions of models?
4. Review desired timeline against OMG Roadmap:
   - Mapping refers to several in-process standards
   - Contribute to/evaluate in-process standards
5. Decide between alternative MDA standards (or to retain the options indicated). Model using real TOGAF data:
   - TEAMS

- Business Architecture?

6. Coordinate with RFP for UML Profile for DODAF/MODAF (SysEng SIG).
7. Use SPEM to model the TOGAF ADM – with MDA mappings to provide a model for the mapping.

References

BMM Business Motivation Metamodel RFC: www.omg.org/cgi-bin/doc?bei/2005-08-03

BPDM Business Process Definition Metamodel RFO: www.omg.org/cgi-bin/doc?bei/03-01-06

BPRI Business Process Runtime Interfaces Platform-Independent Model (PIM) RFP: doc.omg.org/bei/2002-06-08

BSBR Business Semantics of Business Rules RFP: doc.omg.org/br/2003-06-03

CWM Common Warehouse Metamodel (CWM) Specification, Version 1.1: www.omg.org/technology/documents/formal/cwm.htm

EAI UML Profile for Enterprise Application Integration Specification, Version 1.0: www.omg.org/technology/documents/formal/eai.htm

EDOC UML Profile for Enterprise Distributed Object Computing Specification, Version 1.0: www.omg.org/technology/documents/formal/edoc.htm

ITPMF IT Portfolio Management Facility, Adopted Specification: doc.omg.org/dtc/2004-11-03

MOF Meta Object Facility Specification, Version 2.0: www.omg.org/cgi-bin/doc?ptc/2004-10-15

ODM Ontology Definition Metamodel RFP: doc.omg.org/ad/2003-03-40

OMG www.omg.org

OSM Organization Structure Metamodel RFP: doc.omg.org/bei/2004-06-05

PRR Production Rule Representation RFP: doc.omg.org/br/2003-09-03

QVT MOF 2.0 Query/Views/Transformations: doc.omg.org/ad/2002-04-10

RAS Reusable Asset Specification, Version 2.1: www.omg.org/cgi-bin/doc?ptc/2005-04-02

SPEM Software Process Engineering Metamodel, Version 1.1: www.omg.org/technology/documents/formal/spem.htm

UML Unified Modeling Language. This specification is divided into four parts:

- Infrastructure, Adopted Specification: www.omg.org/cgi-bin/doc?ptc/2003-09-15
- Superstructure, Pre-release Version 2.0 (draft formal specification): www.omg.org/cgi-bin/doc?ptc/2004-10-02
- Diagram Interchange, Adopted Specification: www.omg.org/cgi-bin/doc?ptc/2003-09-01
- Object Constraint Language (OCL), Adopted Specification: www.omg.org/cgi-bin/doc?ptc/2003-10-14

XMI XML Metadata Interchange Specification, Pre-release Version 2.1: www.omg.org/cgi-bin/doc?ptc/2004-06-11

About the Integration Consortium

The Integration Consortium is a non-profit, leading industry body responsible for influencing the direction of the integration industry. Its members champion Integration Acumen by establishing standards, guidelines, best practices, research, and the articulation of strategic and measurable business benefits. The Integration Consortium's motto is "Forging Integration Value". The mission is to establish universal seamless integration which engages industry stakeholders from the business and technology community. Further information on the Integration Consortium can be found at www.integrationconsortium.org.

About OMG

The Object Management Group (OMG) is an open membership, not-for-profit consortium that produces and maintains computer industry specifications for interoperable enterprise applications. Its membership includes virtually every large company in the computer industry, and hundreds of smaller ones. Most of the companies that shape enterprise and Internet computing today are represented on the Board of Directors. The OMG flagship specification is the multi-platform Model Driven Architecture (MDA). Further information on OMG can be found at www.omg.org.

About The Open Group

The Open Group is a vendor-neutral and technology-neutral consortium, whose vision of Boundaryless Information Flow™ will enable access to integrated information within and between enterprises based on open standards and global interoperability. The Open Group works with customers, suppliers, consortia, and other standards bodies. Its role is to capture, understand, and address current and emerging requirements, establish policies, and share best practices; to facilitate interoperability, develop consensus, and evolve and integrate specifications and Open Source technologies; to offer a comprehensive set of services to enhance the operational efficiency of consortia; and to operate the industry's premier certification service, including UNIX® system certification. Further information on The Open Group can be found at www.opengroup.org.