



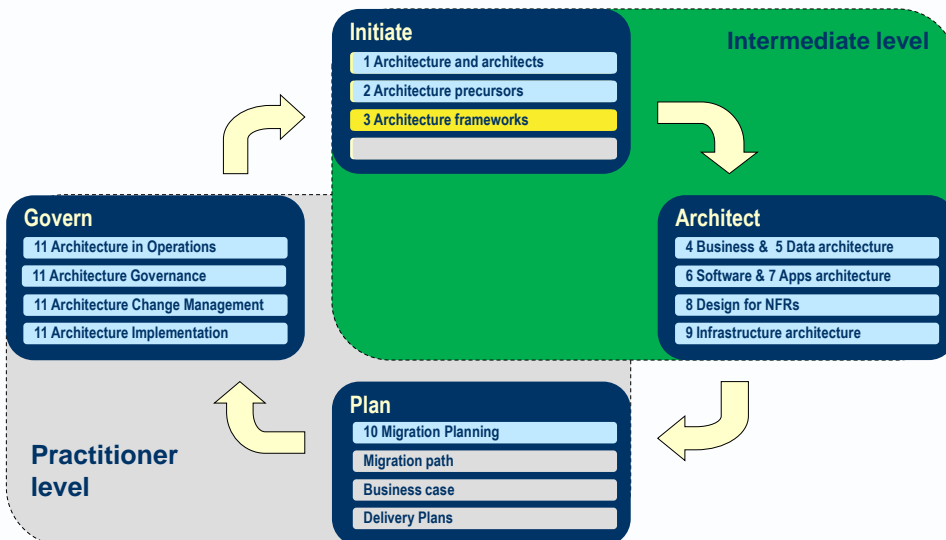
Avancier Reference Model

Architecture Frameworks (ESA 3)

It is illegal to copy, share or show this document
(or other document published at <http://avancier.co.uk>)
without the written permission of the copyright holder

Copyright Avancier Limited

Mapping the reference model to an architecture framework



Copyright Avancier Ltd 2013

3. Architecture frameworks – pass 1



3. Architecture frameworks
Basic concepts
Architecture development processes
Kinds of architecture description
Architecture models and languages
Pre-defined reference models & classifications

Copyright Avancier Limited

What is an architecture framework?



- ▶ Some architecture frameworks are limited to the architectural descriptions of enterprise systems
 - (to conventions, principles and practices for the description of architectures established within a specific domain of application and/or community of stakeholders).
- ▶ But in this reference model, an architecture framework is broader.

Architecture framework	An architecture framework may contain advice on Processes - for architecting Products - for architecture description People - architect roles.
-------------------------------	---

Copyright Avancier Limited

3. Architecture frameworks



3. Architecture frameworks
Basic concepts
Architecture development processes
Kinds of architecture description
Architecture models and languages
Pre-defined reference models & classifications

Copyright Avancier Limited

Architecture processes plan a baseline-to-target migration



Architecture State [an architecture description] at a point in time.
 A baseline architecture describes a system in a state ready to be reviewed and/or revised.
 A target architecture describes a system in a state to be created and implemented in the future.
An intermediate or transitional architecture defines a state of a system between baseline and target.

State	Baseline	Transition 1	Transition 2	Target
Domain	Migration			
Business	Process Organisation Locations	Process Organisation Locations	Process Organisation Locations	Process Organisation Locations
Information Systems	Data Applications	Data Applications	Data Applications	Data Applications
Technology	Infrastructure Technologies	Infrastructure Technologies	Infrastructure Technologies	Infrastructure Technologies

Copyright Avancier Limited

Architecture Processes: ADM



TOGAF [an architecture framework] for enterprise architecture transformation.

It is published on a free-to-read public web site, though its use by a commercial organisation is restricted by copyright conditions.

It is centred on a process called the architecture development method (ADM).

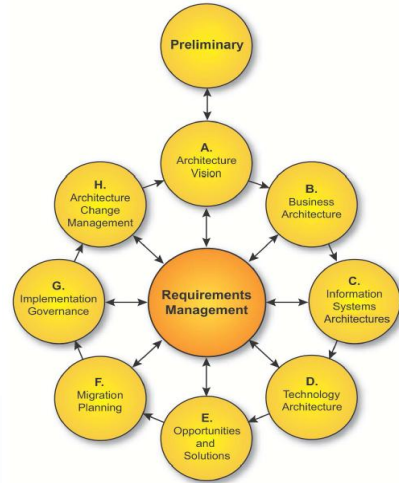


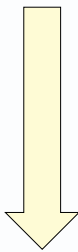
Figure 5-1 Architecture Development Cycle

Copyright Avancier Limited

Architecture Processes: ADM



Architecture Development Method (ADM) [a process] defined TOGAF to develop and use an enterprise architecture. It involves a cycle of 8 phases.



TOGAF's Architecture Development Method (ADM)	Think
Preliminary Phase	establish architecture capability
Phase A: Architecture Vision	feasibility study
Phase B: Business Architecture	
Phase C: IS Architectures	= sum of Data + Apps
Phase D: Technology Architecture	
Phase E: Opportunities & Solutions	eh? what that?
Phase F: Migration Planning	
Phase G: Implementation Governance	not implementation
Phase H: Architecture Change Management	not ITIL
Architecture Requirements Management	continuous

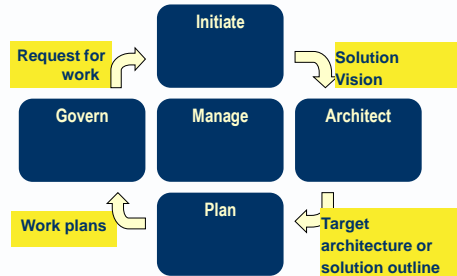
Copyright Avancier Limited

Architecture Processes: ADM



**AM:
Avancier
Methods**

[an architecture framework] focused on solution architecture, though it also addresses enterprise architecture rationalisation. It is published on a free-to-read public web site, though its use by an organisation is limited by copyright conditions. It features a solution architecture process with four phases (Initiate, Architect, Plan and Govern), each subdivided into lower level processes.



Copyright Avancier Limited


3. Architecture frameworks - pass 1



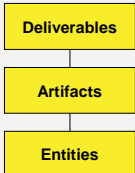
3. Architecture frameworks
Basic concepts
Architecture development processes
Kinds of architecture description
Architecture models and languages
Pre-defined reference models & classifications

Copyright Avancier Limited

Architecture description



Architecture content framework




[a passive structure] for organising an architecture description composed of deliverables, artefacts and entities.

Deliverables contain artifacts, which are in turn composed from architectural entities.

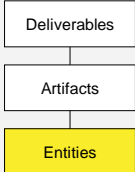
Conversely: one entity can appear in several artifacts, which can appear in several deliverables.

Copyright Avancier Limited

Architecture description



Architectural entity



A discrete architectural element, an object in an architecture repository, that is reusable in different artifacts, and is definable using a standard template. An entity instance may be decomposed into finer-grained instances of the same type.

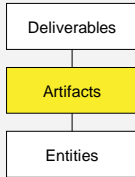
E.g.
Business goal/objective
Business service
Business function, Business process step
Role, Actor
Location
Data store, Data flow,
Application, Use case,.
Technology, Device, Network

Copyright Avancier Limited

Architecture description



Architecture artifact

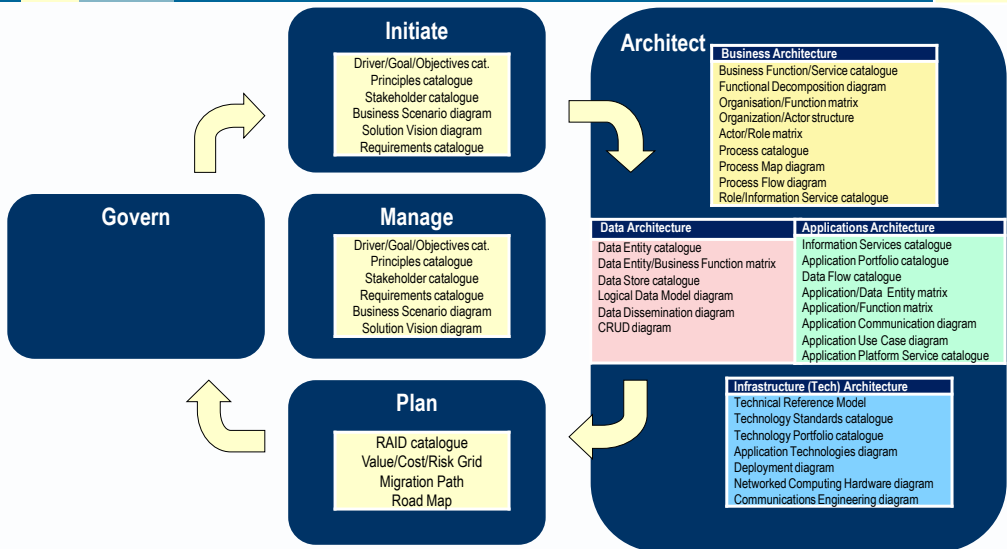


[a model] that conforms to one of three artefact types: catalogue, matrix and diagram.

E.g.
 Requirements traceability matrix
 Business process model, Organisation structure,
 Data model, Data Entity-Business Function matrix,
 Data Flow Diagram, Sequence Diagram
 Hardware configuration diagram.

Copyright Avancier Limited

Artefacts mapped to process

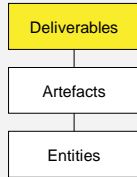


Copyright Avancier Limited

Architecture Deliverables



Architecture deliverable



[a document] that architects produce or contribute to, for approval by sponsors if not all stakeholders. It should conform to a document type defined by a standard contents list.

An architecture description deliverable contains architecture artefacts or models.

- ▶ Request for Work,
- ▶ Statement of Work,
- ▶ Architecture Requirements,
- ▶ Architecture Definition,
- ▶ RAID Catalogue,
- ▶ Migration Plan.

Copyright Avancier Limited

3. Architecture frameworks



3. Architecture frameworks
Basic concepts
Architecture development processes
Kinds of architecture description
Architecture models and languages
Pre-defined reference models & classifications

E.g. APQC Process classification

E.g. Zachman Framework

Copyright Avancier Limited

A classification of architecture description

“Columns show “the primitive interrogatives”

Zachman Framework v3		What	How	Where	Who	When	Why
Idealisation	Stakeholder perspective	Inventory sets	Process flows	Distribution networks	Responsibility assignments	Timing cycles	Motivation intentions
Scope Contexts	Executive						
Business Concepts	Business management						Business goals
System Logic	Architect			Network diagram			
Technology Physics	Engineer						
Tool components	Technician						
Operations - Instance classes	Enterprise						

“Rows show “reification - the transformation of an abstract idea into an instantiation... labeled
 • Identification,
 • Definition,
 • Representation,
 • Specification,
 • Configuration &
 • Instantiation.”

Copyright Avancier Limited

A reference model for architecture description



APQC’s Process Classification Framework (PCF).



This standard hierarchical classification of the functions in a commercial enterprise can provide you with a means to

- Structure baseline activities
- Identify and structure required activities.

Copyright Avancier Limited

APQC updated and limited to 3 levels



1. UNDERSTAND MARKETS AND CUSTOMERS		
1.1 Determine customer needs and wants		
1.1.1 Conduct qualitative assessments		
1.1.1.1 Conduct customer interviews		
1.1.1.2 Conduct focus groups		
1.1.2 Conduct quantitative assessments		
1.1.2.1 Develop and implement surveys		
1.1.3 Predict customer purchasing behavior		
1.2 Measure customer satisfaction		
1.2.1 Monitor satisfaction with products and services		
1.2.2 Monitor satisfaction with complain		
1.2.3 Monitor satisfaction with commun		
1.3 Monitor changes in market or customer expecte		
1.3.1 Determine weaknesses of produc		
1.3.2 Identify new innovations that mee		
1.3.3 Determine customer reactions to		
	2. DEVELOP VISION AND STRATEGY	
	2.1 Monitor the external environment	
	2.1.1 Analyze and understand competition	
	2.1.2 Identify economic trends	
	2.1.3 Identify political and regulatory issues	
	2.1.4 Assess new technology innovations	
	2.1.5 Understand demographics	
	2.1.6 Identify social and cultural changes	
	2.1.7 Understand ecological concerns	
	2.2 Define the business concept and organizational strategy	
	2.2.1 Select relevant markets	
	2.2.2 Develop long-term vision	
	2.2.3 Formulate business unit strategy	
	2.2.4 Develop overall mission statement	
	2.3 Design the organizational structure and relationships between organizational units	
	2.4 Develop and set organizational goals	

Copyright Avancier Limited

3. Architecture frameworks – end of pass 1



▶ **SHOW RELEVANT MOCK EXAM QUESTIONS**

Copyright Avancier Limited

3. Architecture frameworks – pass 2b



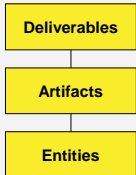
3. Architecture frameworks
Basic concepts
Architecture development processes
Kinds of architecture description
Architecture models and languages
Pre-defined classifications & reference models

Copyright Avancier Limited

Architecture description



Architecture content framework



[a passive structure] for organising an architecture description composed of deliverables, artefacts and entities.

Deliverables contain artifacts, which are in turn composed from architectural entities.

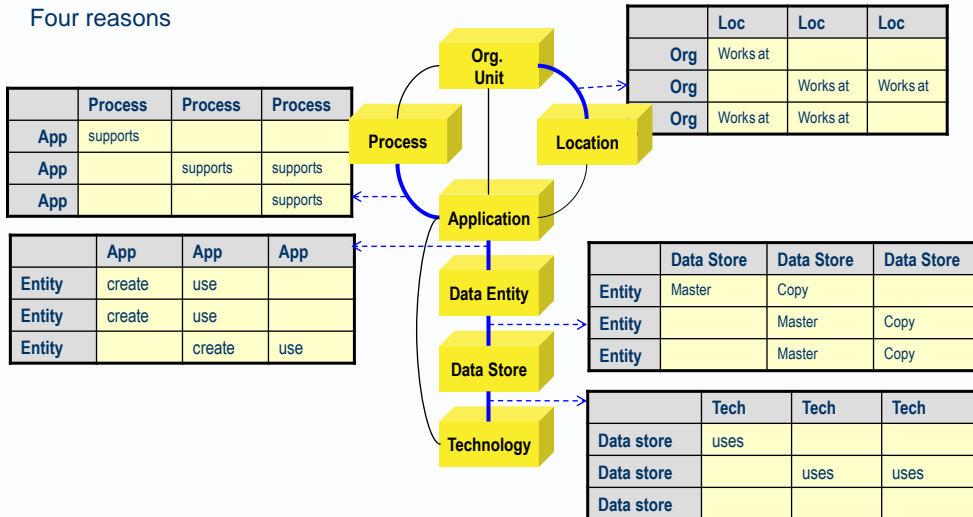
Conversely: one entity can appear in several artifacts, which can appear in several deliverables.

Copyright Avancier Limited

Why draw *mapping tables* between architectural entities?



Four reasons



Copyright Avancier Limited

Architecture description: mappings



Mapping

[a correspondence] that is drawn between elements of the same or different structures.

Correspondences can be mapped for several purposes including:

- ▶ gap analysis (section 10)
- ▶ impact analysis (section 11),
- ▶ requirements traceability analysis
- ▶ cluster analysis (section 4).

E.g.

- ▶ **organisation unit to business function,**
- ▶ business function to application,
- ▶ application to platform technology.
- ▶ data entity to business function,
- ▶ data entity to data store,
- ▶ data entity to data quality.

	Function	Function
Org	Performs	
Org		Performs
Org	Performs	Performs

Copyright Avancier Limited

Four reasons to draw *mapping tables* between architectural entities



- ▶ Gap analysis
 - To find potentially missing items
- ▶ Traceability analysis
 - To check deliverables meet goals and solutions solve problems.
- ▶ Impact/Dependency analysis
 - To find the effects of a change
- ▶ Cluster analysis
 - To group closely-coupled items into encapsulated components

Find this in your manual

Copyright Avancier Limited

Mappings: Inter-state gaps

Gap analysis
 Traceability analysis
 Impact/Dependency analysis
 Cluster analysis



- ▶ You may want to compare
 - a vision with reality
 - a specification with an implemented solution
 - a baseline with a target

Target Apps Baseline Apps	Billing	CRM	Business Intelligence	Baseline
Billing	Port to new platform			
CRM		Leave as is		
Resourcing			???	Decommission or reconsider
Target			Buy or build	

Copyright Avancier Limited

Mappings for traceability analysis

Gap analysis
Traceability analysis
 Impact/Dependency analysis
 Cluster analysis



► Traceability analysis means following mappings to check deliverables meet goals and solutions solve problems.

Solution items	Business Intelligence	Sales mobile device	CRM	Requirements with no solution
Requirements				
Faster Ordering		Satisfied by		
Remote Working		Satisfied by		
Better Forecasts	Satisfied by			
Lower Sale Cost			???	No solution
Solution items with no requirement			No requirement	

- E.g. 40 solution elements
- 800 functional requirements (hierarchy) + 200 NFRs!

Mappings for impact/dependency analysis

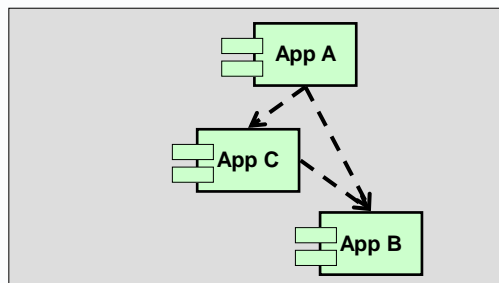
Gap analysis
 Traceability analysis
Impact/Dependency analysis
 Cluster analysis



► Impact analysis implies recording dependencies, so you can follow mappings to check if a change to one item causes a change to related items.

	App A	App B	App C
App A		Depends on	Depends on
App B			
App C		Depends on	

► The same thing in UML



Use cluster analysis to group activities acting on the same data

Gap analysis
Traceability analysis
Impact/Dependency analysis
Cluster analysis



- ▶ Map actor or activity to data element in a matrix

Actor or Activity Data element	Billing	Delivery	Sales	Reporting
Customer	Use	Use	Create	Use
Order	Use	Use	Create	Use
Delivery	Use	Create		Use
Invoice	Use	Create		Use
Payment	Create			Use
Report				Create

- ▶ Use the NW corner method to sort rows and columns

- ▶ Cluster on “create” to suggest the boundaries of applications and data stores

Actor or Activity Data element	Sales	Delivery	Billing	Reporting
Customer	Create	Use	Use	Use
Order	Create	Use	Use	Use
Delivery		Create	Use	Use
Invoice		Create	Use	Use
Payment			Create	Use
Report				Create

Copyright Avancier Limited

ISO 42010 (ex ANSI 1471)



- ▶ What are the key concepts in ISO 42010?

ISO/IEC 42010

Recommended Practice for Architecture Description of Software-Intensive Systems.

A standard for software architecture or system architecture.

It focuses on the description of an architecture as the concrete artifact representing the abstraction that is software architecture or system architecture.

Commonly known by its original identity ANSI 1471.

Copyright Avancier Limited

Architecture viewpoints and views



Viewpoint [a work product description] that defines the conventions for constructing, interpreting and using views to frame specific concerns about a system.
 A view point template records:

- what – the name of the viewpoint
- why - concern(s) that the viewpoint addresses
- who - stakeholder(s) who have the concerns
- how - model kind(s) used in the view.

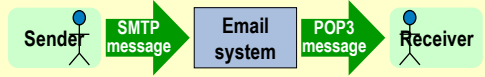
View [a work product] that shows a part or slice of an architecture description, to address the particular concerns of particular system stakeholders.

It is drawn according to the conventions of one viewpoint.
 It may contain one or more models.

Template or type

What	Why?	Who cares?	How to draw it?
Context diagram	The scope of the system	Owners and designers	

Example or instance



Copyright Avancier Ltd 2013

An enterprise's viewpoint library



Even a small and simple enterprise architecture description is difficult to grasp all at once. You may need many viewpoints.

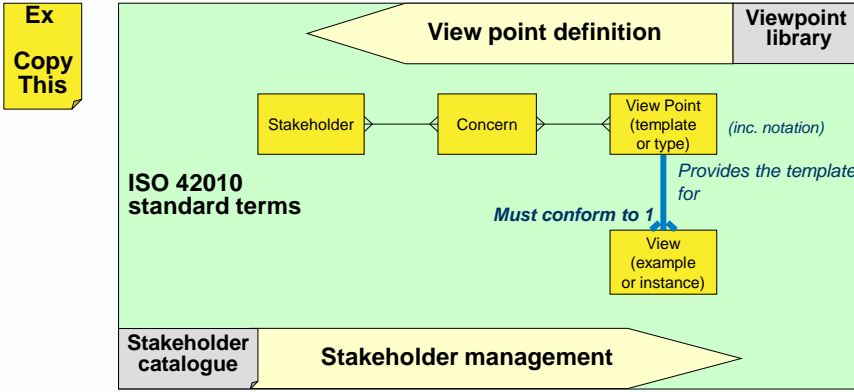
What	Why	Who	How
Viewpoint	Addresses these concerns	Of these stakeholders	Using this kind of model
Business distribution	Where we do our business	CEO COO	Hierarchical tree showing locations of functions
Location activity	What we do at each location	Facilities Manager	
Deployment overview	Where our kit is	IT Services Manager	
Data model	Data structure Data item types Potential access paths	Systems analysts Database designers Domain experts	IDEF1X standard
Application usage	Which applications used by which roles in which business functions	Application portfolio manager Systems analysts	In-house template

Copyright Avancier Limited

The simple version implied by BCS reference model



- ▶ BCS exam may test your understanding of the relationships between these four essential concepts in ISO 42010 (aka ANSI 1471)



Copyright Avancier Limited

ISO 42010: "The architecture of a system



"fundamental concepts or properties of a system in its environment embodied in its **elements**, relationships, and in the principles of its design and evolution"

- ▶ Note: the elements include components and processes

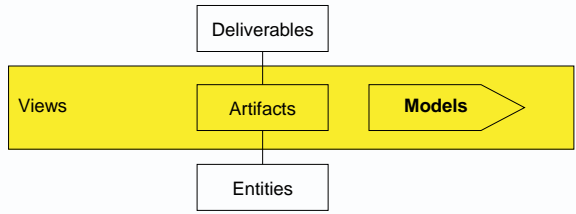
A well-known definition in the ISO standard, you may come across it elsewhere

Copyright Avancier Limited

3. Architecture frameworks



3. Architecture frameworks
Basic concepts
Architecture development processes
Kinds of architecture description
Architecture models and languages
Pre-defined classifications & reference models

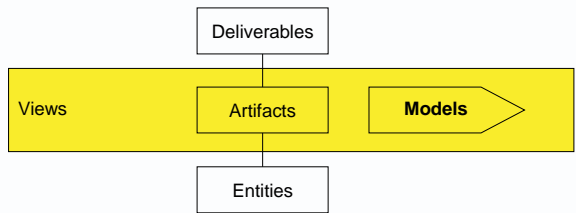


Copyright Avancier Limited

Architecture description: models and abstractions



Model [a description] a thing or description that simplifies or **abstracts from** another thing or description.
It displays or records some properties of what is modelled and enables questions to about it to be answered.



Copyright Avancier Limited

Abstraction and its opposites



Abstraction	[a technique] by which a simpler description is derived from other descriptions or from reality.
Refinement	[a technique] that yields a detailed description that conforms to a more abstract description. Everything in the abstraction holds, perhaps in a somewhat different form, in the refinement.
Concretion	[a technique] that instantiates a description as one or more real, active (or activatable) components.

Copyright Avancier Limited 2013


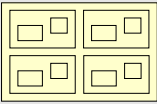
EA is more *abstract* than SA



Omission	Composition packing details into boxes	Generalisation removing differences of detail	Idealisation removing differences between physical forms
	Coarse-grained views and models	Common components and processes	Conceptual/logical views and models
Elaboration	Decomposition	Specialisation	Realisation

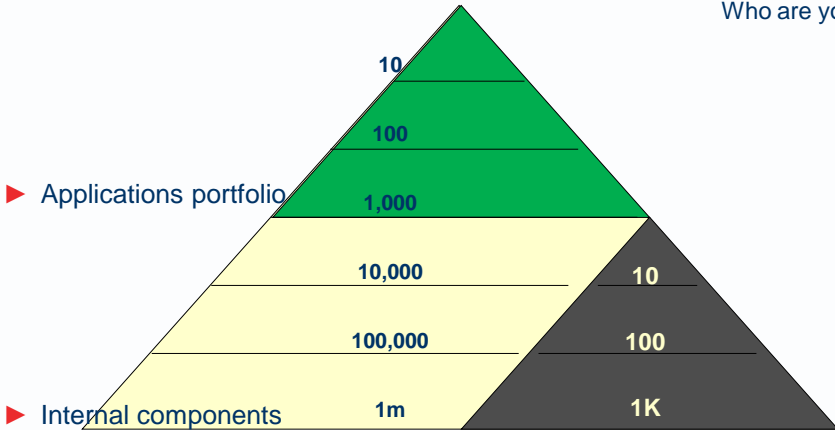
Copyright Avancier Limited 2013

Comp-decomp Avancier

<p>Composition</p> 	<p>[a technique] that assembles parts into a whole and/or hides components behind a façade.</p> <p>Architects describe systems in terms of coarse-grained components, processes and services.</p>
<p>Decomposition</p> 	<p>[a technique] that divides a whole into parts and/or identifies components behind a façade.</p> <p>The conventional advice is that it is difficult to maintain the integrity of a hierarchical structure that is decomposed more the three or four levels (or more than a thousand elements) from the top.</p>

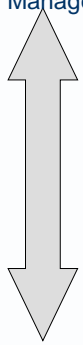
Copyright Avancier Limited 2013

Composition – Coarse-grained views and models Avancier



Who are you talking to?

Manager



Technician

Copyright Avancier Limited 2013

Gen-spec



Generalisation	[a technique] that defines properties shared by subtypes or entities. Architects look to maximise re-use of common components, processes and services across the enterprise.
Specialisation	[a technique] that extends or modifies generic properties to define a subtype or smaller population of entities. It can mean configuring an instance by selecting variable values from a general range.

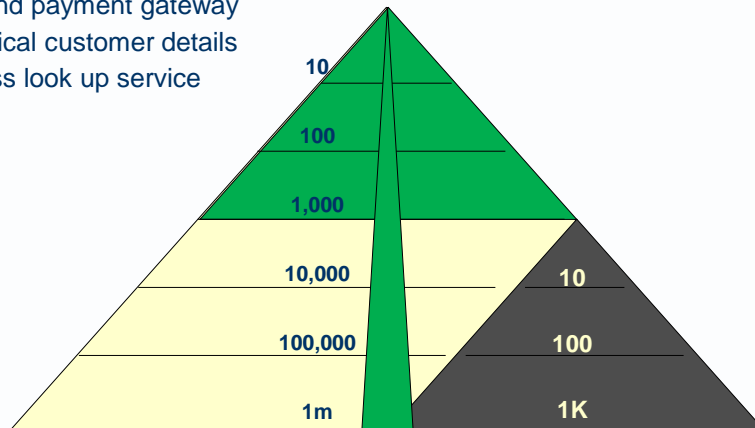
- ▶ Beware that generalisation of
 - architecture descriptions can yield abstractions of low practical benefit.
 - solution components for 'flexibility' can lead to performance problems.

Copyright Avancier Limited 2013


Generalisation - Common components and processes



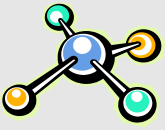
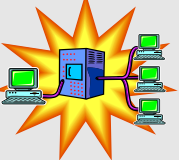
- ▶ E.g.
- ▶ Single sign on, across 1,000 applications
- ▶ In-bound payment gateway
- ▶ Canonical customer details
- ▶ Address look up service




Copyright Avancier Limited 2013


Avancier

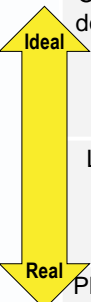
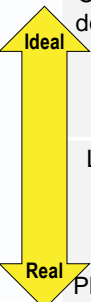

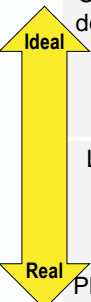

Ideal-real

<p>Idealisation</p> 	<p>[a technique] that reverse-engineers a more logical description by omitting some details relevant to a particular physical form of the thing described. Architects produce and work with logical descriptions of components, processes and services.</p>
<p>Realisation</p> 	<p>[a technique] that forward-engineers a more physical description. It adds details relevant to a particular physical form of the thing described.</p> <p>OR concretion: the instantiation of a description as one or more active, run-time, components.</p>

Copyright Avancier Limited 2013


Avancier

The idealisation or model hierarchy

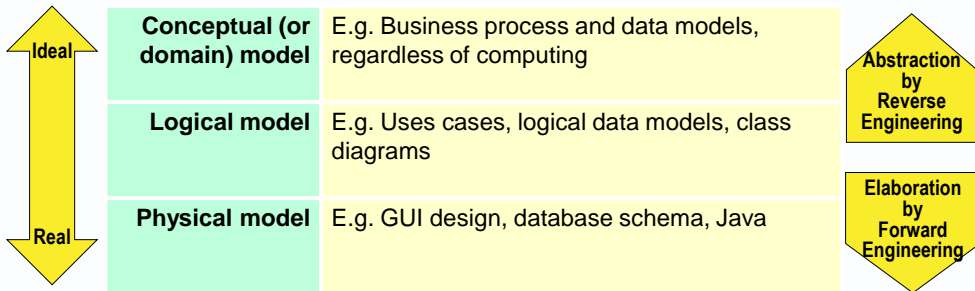
Idealisation hierarchy	The classic hierarchy of conceptual, logical and physical model.	
<div style="display: flex; align-items: center;">  <div style="background-color: #e0e0e0; padding: 5px;">Conceptual (or domain) model</div> </div>	[an artefact] an abstract logical model that defines terms and concepts in a business or problem domain without reference to any computer application.	
<div style="display: flex; align-items: center;">  <div style="background-color: #e0e0e0; padding: 5px;">Logical model</div> </div>	[an artefact] a model of a particular system that excludes details of that system's physical implementation.	
<div style="display: flex; align-items: center;">  <div style="background-color: #e0e0e0; padding: 5px;">Physical model</div> </div>	[an artefact] a model of a particular system that is vendor or technology specific and/or includes details of its physical implementation.	

Copyright Avancier Limited

MDE: Model-Driven Engineering



MDE: Model-Driven Engineering [a technique] used in methods and tools for transforming a conceptual model to a logical model, and a logical model to a physical model, and the reverse.
It covers forward engineering and reverse engineering.

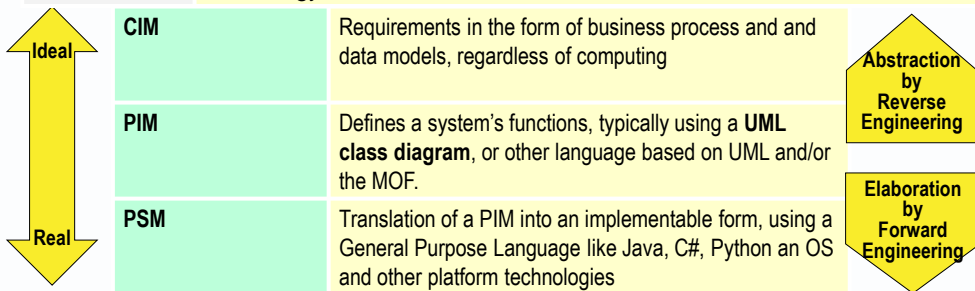


Copyright Avancier Limited

Architecture description: models and abstractions



Model-Driven Architecture (MDA) a vision of the Object Management Group (OMG) that encourages vendors to develop tools for Model-Driven Engineering to standards defined by the OMG. The MDA idealisation hierarchy is:
 •computation-independent model (CIM),
 •platform-independent model (PIM), unrelated to a specific technology
 •platform-specific model (PSM), related to specific infrastructure technology.



Copyright Avancier Limited

System modeling techniques



Modelling language

[a standard] that defines shapes for representing architecture entities and arc/line styles for representing relationships between them.

Three international varieties are IDEF, UML and ArchiMate.

Models commonly used by architects include

- process models,
- data models,
- context diagrams,
- use case diagrams,
- data flow diagrams, and
- interaction/sequence diagrams.

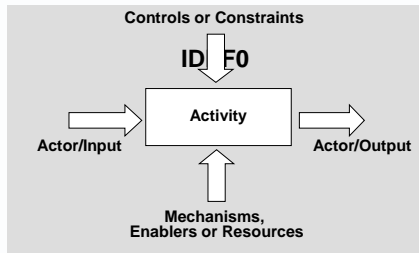
Copyright Avancier Limited

Integration Definition language (IDEF)

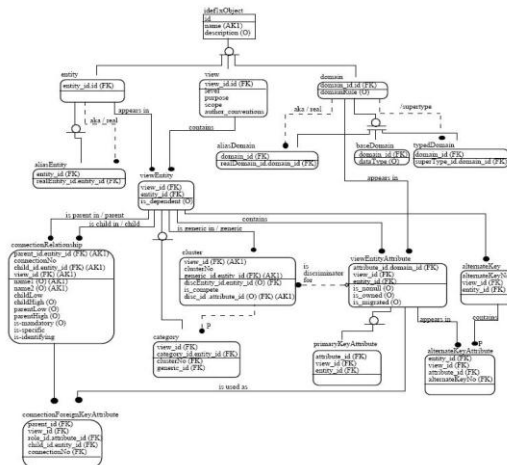


- ▶ Grew out of 1970s USAF standards, best known for

- ▶ IDEF 0 Function Models



- ▶ IDEF 1X Data Models (Wikipedia)

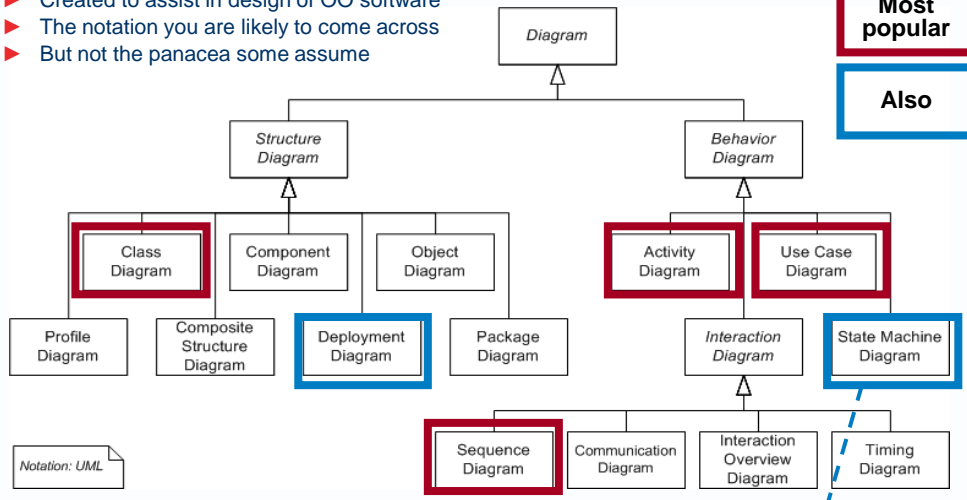


Copyright Avancier Limited

UML (OMG standard)



- ▶ Created to assist in design of OO software
- ▶ The notation you are likely to come across
- ▶ But not the panacea some assume



- ▶ Sessions (short term)
- ▶ Data entities (long term)

Copyright Avancier Limited

Avancier Methods core framework for EA with ArchiMate



	<i>Passive Structure acted upon</i>	<i>Creates or uses</i>	<i>Required Behaviour event to result</i>	<i>Assigned to</i>	<i>Logical Structure groups activities</i>	<i>Realised by</i>	<i>Physical Structure performs activities</i>
Business		Business Service	Business Process	Function	Business Role	Organisation Unit	Actor
Information Systems	Data Entity	IS Service		Application Interface		Application	
Infrastructure Technology		Platform Service		Platform Interface		Platform Application	Device

Copyright Avancier Limited 2008 to 2015

Architectural modelling languages



Integration DEFinition (IDEF) language	[a modelling language] in the field of systems and software engineering, originally funded by the US DoD. Its most-well-known notations are IDEF0 (a process modeling language building on SADT) and IDEF1X for information models and database design.
Unified Modelling Language (UML)	[a modelling language] maintained by the Object Management Group. It was designed to help in object-oriented software design, though often used outside of that. It includes structural models such as class diagrams and deployment diagrams, and behavioural models such as use case, activity and sequence diagrams.
ArchiMate	[a modelling language] maintained by the Open Group. It was designed to help in architecture description. Components, interfaces and services are shown in distinct boxes. It overlaps with UML but is intended for more abstract design.

Remember their names

Copyright Avancier Limited

3. Architecture frameworks



3. Architecture frameworks
Basic concepts
Architecture development processes
Kinds of architecture description
Architecture models and languages
Pre-defined classifications & reference models

Copyright Avancier Limited

Pre-defined classifications and reference models



Reference model

- FEA for US federal government (3)
- APQC - generic business (4)
- BIAN for banking (4)
- TMF for telecoms
 - eTOM – Business Architecture (4)
 - SID – Data Architecture
 - TAM – Applications Architecture
- SCOR for supply-chain businesses (4)
- ProAct for retailers (4)
- Industry canonical data models (5)
- TOGAF's III-RM (7)
- TOGAF's TRM (9)

[a pattern] an abstract structure or classification used to create more specific models.

It can be a structure of components, processes or data elements.

It is sometimes applicable to a particular industry or business domain.

Can act as a design pattern

Copyright Avancier Limited

The core of the enterprise continuum



Architecture repository

[a data store] an information base used by architects; a system that holds and manages all the meta data that describes an enterprise and its information systems.

Its structure is defined in some kind of schema or architecture meta model.

The content of the repository can be categorised using the Zachman Framework or Enterprise Continuum.

Copyright Avancier Limited

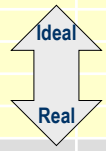
Pre-defined classifications and reference models



Zachman framework [a pattern] “A logical structure for classifying and organising the descriptive representations of an Enterprise that are significant to managers and to developers of Enterprise systems.”

- ▶ A classification scheme for reusable architecture assets
- ▶ Nothing more or less than a set of pigeon holes for architecture description artefacts

Zachman Framework	What	How	Where	Who	When	Why
Scope Contexts						
Business Concepts						
System Logic						
Technology Physics						
Tool components						
Operations – Instance classes						



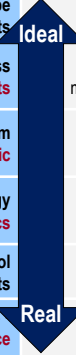
Copyright Avancier Limited

Remember the meanings of rows and columns

2011, essence of the Zachman Framework version 3



Zachman Framework v3		What	How	Where	Who	When	Why
Idealisation	Stakeholder perspective	Inventory sets	Process flows	Distribution networks	Responsibility assignments	Timing cycles	Motivation intentions
Scope Contexts	Executive						
Business Concepts	Business management						
System Logic	Architect						
Technology Physics	Engineer						
Tool components	Technician						
Operations – Instance classes	Enterprise						



Copyright Avancier Limited

The BCS reference model entry says



Zachman framework

“A logical structure for classifying and organising the descriptive representations of an Enterprise that are significant to managers and to developers of Enterprise systems.”

Drawn as table or grid:

The 6 columns are primarily analysis questions. But they are also interpreted as architecture domains (entities?) (data, process, network etc.).

The 6 rows are primarily levels of **idealisation-realisation** from highest level context to operational systems, but they are also interpreted as **stakeholder groups** and **architecture viewpoints**.

Zachman says the rows should not be interpreted as levels of decomposition.

Copyright Avancier Limited

Pre-defined classifications and reference models



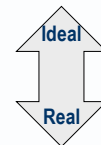
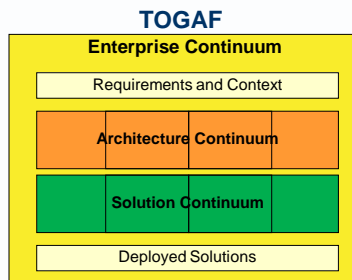
Enterprise continuum

[a pattern] a logical structure for classifying and organising architecture description artefacts.

It is a core part of TOGAF.

It can be drawn as a table or grid; from top to bottom is ideal to real; from left to right is general to specific.

- ▶ A classification scheme for reusable architecture assets
- ▶ Nothing more or less than a set of pigeon holes for architecture description artefacts

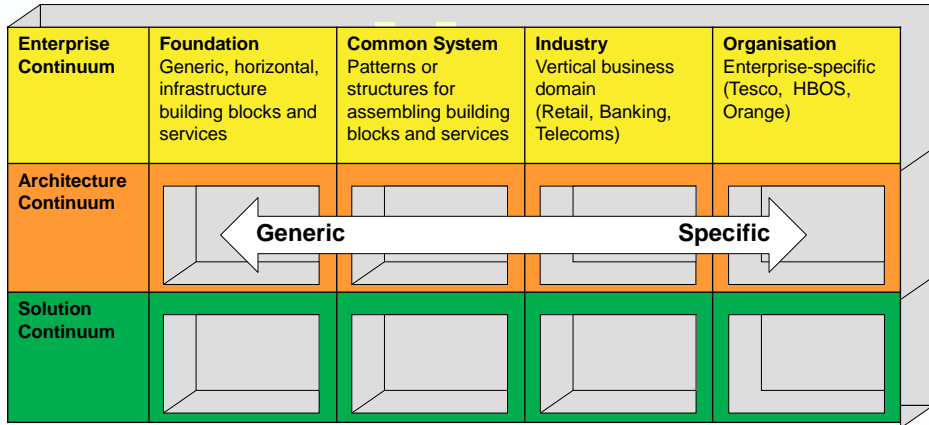


Copyright Avancier Limited

TOGAF's Enterprise Continuum – left to right



- ▶ From generic foundations, nuts, bolts and building blocks
- ▶ To organisation-specific structures and components



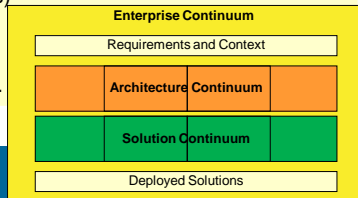
Copyright Avancier Limited

Architecture description frameworks



Enterprise continuum

A structure for architecture description documentation used in TOGAF.
 A classification scheme for the contents of an architecture repository.
 Drawn as a table or grid, the rows are similar to those in the Zachman Framework. The columns are a spectrum from universal to unique.
The 4 columns represent generalisation-specialisation, ranging from universal to bespoke or uniquely configured.
 Foundation: Structures and items that are universal.
 Common systems: Structures (composed of foundation items) that are used across most business domains.
 Industry: Structures and items used by enterprises in one business domain (say Telecoms or Banking).
 Organisation: Structures and items specific or bespoke to a single enterprise.
The 4 rows represent levels of idealisation. From top to bottom:
 Requirements and context (architecture precursors)
 The architecture continuum (see below)
 The solution continuum (see below)
 Deployed solutions (architecture implementations).



Copyright Avancier Limited

The core of the enterprise continuum

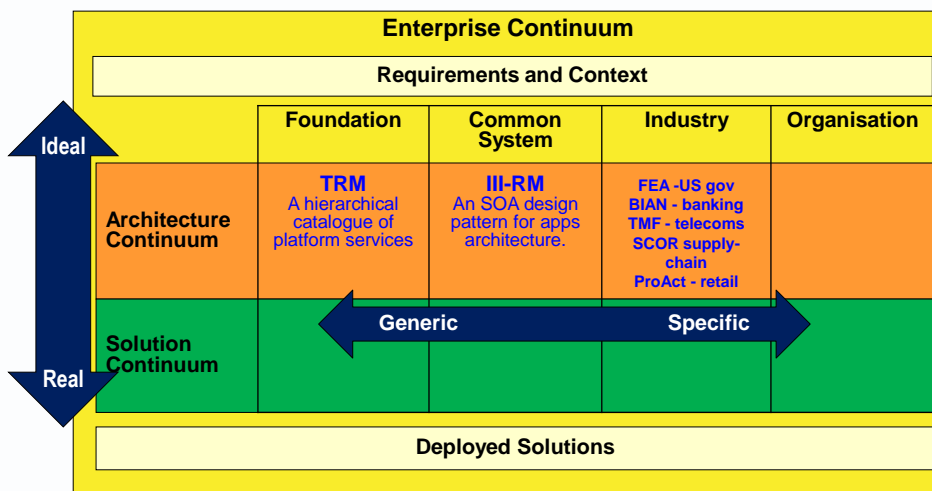


Architecture continuum	A higher-level spectrum in the enterprise continuum, which contains logical or vendor-neutral specifications of requirements. It corresponds to the logical model level of the idealisation hierarchy.
Solutions continuum	A lower-level spectrum in the enterprise continuum, which contains specifications of products and services that implement the logical specifications. It corresponds to the physical model level of the idealisation hierarchy.

	Foundation	Common System	Industry	Organisation
	Generic, horizontal, infrastructure building blocks and services	Patterns or structures for assembling building blocks and services	Business domain/vertical (Retail, Banking, Telecoms)	Enterprise-specific (Tesco, HBOS, Orange)
Architecture continuum	e.g. TRM (Technical Reference Model) SIB (Standards Information Base)	e.g. application integration patterns such as the III-RM.	e.g. Function, Data and Process models	e.g. Bespoke application specifications
Solution continuum	Strategic products Operating systems	Product assemblies (Email system, Security system)	e.g. COTS Packages	Bespoke solutions

Copyright Avancier Limited

TOGAF's Enterprise Continuum – Reference Models



Copyright Avancier Limited