WIT 2016 ITA Module

Architecture Modeling
Lecture Group #2 - Part 3
Lecture Group #2 - Part 3
Architecture Modeling

Design Techniques - Modeling with ArchiMate® 2.1
http://pubs.opengroup.org/architecture/archimate2-doc/
Design Techniques
Modeling with ArchiMate® 2.1

http://pubs.opengroup.org/architecture/archimate2-doc/
About “Expressiveness”
...and modeling ideas elegantly
Example: City Infrastructure
Example: City Infrastructure

Example: Subway modeling (continued)

Example: City Infrastructure
Subway modeling?
Example: Subway modeling (continued)
Story telling Example: Feeding Oscar

1. Phil / Sharon
2. 6 a.m.
3. Feeding
4. Oscar
5. Get fed
6. Provide food
7. Hunger
8. 6 a.m.
Example: Feeding Oscar (continued)
Example: Talented Captain Hook
Definition

A modeling standard published by The Open Group in 2009 (v1.0)/2012 (v2.1). A design technique for describing architectures. Presents a clear set of concepts within architecture domains, offers a uniform structure for describing the contents of each domain.

...each domain is specified by a meta-model, constraining the diagrams that can be created, and allowing consistency of notation and re-use of concept elements between Views.

Presents a clear set of concepts to establish RELATIONSHIPS (.e. MAP) between domains.

...allows the connection of models belonging to different layers (i.e. Business/Application/Data/Technology), hence helping an Architect to document View consistency.
Purpose

The purpose of ArchiMate® 2.1 is to:
- be semantically precise so to help analysing and visualising relationships among problem/solution domains in an unambiguous way,
- be read by all stakeholders, and convey one same meaning,
- be intuitive to understand, with little or no training to understand the models.
To transcend Diagramming/Notational Silos
About UML Extensions

The Object Management Group Unified Modeling Language (UML) can be extended to support Enterprise Architecture modeling.

The Enterprise Architecture EXTENSION Profile (UML EAP) proposes to combine:
- however it does not include the Object Management Group Business Motivation Modeling (OMG BMM [http://www.omg.org/spec/BMM/]).
About Archimate Extensions

Model Extensions introduced in 2012:
- Motivation Extension
- Implementation / Migration Extension (to picture current/target state increments, for program planning)

Archimate Version 3.0 being finalized in 2016.
Modeling Notation

Modeling Elements Color-schemes

Model Element Relations & Grammar

Model Element Types & Layers

Modeling Notation
# Core Elements Classification by Layer

<table>
<thead>
<tr>
<th>Layer</th>
<th>Structural Concepts</th>
<th>Behavioral Concepts</th>
<th>Informational Concepts</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Business</strong></td>
<td>Business actor</td>
<td>Business role</td>
<td>Business process</td>
</tr>
<tr>
<td></td>
<td>Business collaboration</td>
<td>Business interface</td>
<td>Business function</td>
</tr>
<tr>
<td></td>
<td>Location</td>
<td>Business object</td>
<td>Business interaction</td>
</tr>
<tr>
<td><strong>Application</strong></td>
<td>Application component</td>
<td>Application collaboration</td>
<td>Application function</td>
</tr>
<tr>
<td></td>
<td>Application interface</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Technology</strong></td>
<td>Node</td>
<td>Device</td>
<td>Infrastructure function</td>
</tr>
<tr>
<td></td>
<td>Network</td>
<td>System software</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Communication path</td>
<td>Infrastructure interface</td>
<td></td>
</tr>
</tbody>
</table>
Core Semantics (Who, Acts, on What, How)
Core Semantics (Who, Acts, on What, How)

The language consists of 3 main TYPES of core elements:
- ACTIVE structure elements are entities capable of performing behavior (business actors, application components, devices, etc.)
- BEHAVIOR elements are units of activity performed by one or several active structure elements (processes, interactions, use cases, etc.)
- PASSIVE structure elements are objects on which behavior is performed, such as data (business entities, data objects, etc.)

Note: External View model elements EXPOSE model elements from Internal View.
ACTIVE Model Elements

An Entity of your design capable of performing behavior (i.e. Subject).
A permanent or temporary grouping (or aggregation) of two or more structure elements, working together to perform some collective behavior (i.e. Collaboration).
**BEHAVIORAL Elements**

A unit of activity performed by one or more active structural elements (i.e. Verb).

A unit of behavior performed by a collaboration of two or more structure elements (i.e. Interaction).
PASSIVE Elements

An object on which behavior is performed, usually information or data objects.
Symbols & Relationships
Symbols & Relationships

Additional Relationships

Behavioral Relationships

Structural Relationships

Symbols & Relationships
Structural Relationships

Structural relationships model the STRUCTURAL COHERENCE of concepts of same (or different) model element types.

- **Composition**: any object is composed of 1 or more other objects,
- **Aggregation**: groups a number of other objects,
- **Assignment**: links active elements with units of behavior that are performed by them, links business actors with business roles that're fulfilled by them,
- **Realization**: links logical entity with a more concrete entity that realizes it,
- **Used by**: models use of service by \{processes | functions | interactions\}, access to interfaces by \{roles | components | collaborations\},
- **Access**: models access of behavioral concepts to \{business | data\} objects (directional if arrowhead),
- **Association**: models other relationships between objects.
Behavioral Relationships

Behavioral relationships model DYNAMIC DEPENDENCIES between behavioral concepts.

**Triggering:**
- describes temporal or causal relationships between processes, functions, interactions and events
- no distinction between active triggering or passive causal relationship

**Flow:**
- describes exchange/transfer of information/value between processes, functions, interactions, events
- does not imply temporal or causal relationship
Additional Relationships

**Grouping**: objects belong together based on some common characteristics,

**Junction**: used to connect dynamic relationships of the same type,

**Specialization**: indicates that an object is a specialization of another object,

**Derived relationships**: two relationships that join at an intermediate element can be combined & replaced by the weaker of the two.
Notations introduced in v2.0

The **Motivation Extension** adds concepts such as goal, principle, and requirement. It addresses the way the enterprise architecture is aligned to its context, as described by motivational elements.

The **Implementation and Migration Extension** adds concepts to support the later ADM phases, related to the implementation and migration of architectures.

### Motivation Extension

- **Stakeholder**
- **Assessment**
- **Driver**
- **Goal**
- **Requirement**
- **Constraint**
- **Principle**

### Implementation and Migration Extension

- **Work package**
- **Deliverable**
- **Plateau**
- **Gap**
Modeling Elements Color-schemes

Either of two: (1.) Layered coloring, or (2.) Grammar coloring.

1. "Layered" color-scheme (default in modeling tools):
   - Yellow-for-Business layer
   - Blue-for-Application layer
   - Green-for-Infrastructure layer

2. "Grammar" color-scheme (best practice of archimate modeling):
   - Blue-for-actors
   - Yellow-for-behavior
   - Green-for-acted-upon
Archimate Layers

Rows 1/2/3

Rows 3/4/5

Rows 5/6
Mapping Archimate to Rows 1/2/3
BUSINESS / INFORMATION Layer

The Business layer offers products and services to external customers, which are realised in the organisation by business processes performed by business actors.

In the Business Layer, data is represented as information (i.e. data in context).
# Business/Information Layer Concepts (1/2)

<table>
<thead>
<tr>
<th>Concept</th>
<th>Description</th>
<th>Notation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business actor</td>
<td>An organizational entity that is capable of performing behavior.</td>
<td><img src="image" alt="Business actor" /></td>
</tr>
<tr>
<td>Business role</td>
<td>The responsibility for performing specific behavior, to which an actor can be assigned.</td>
<td><img src="image" alt="Business role" /></td>
</tr>
<tr>
<td>Business collaboration</td>
<td>An aggregate of two or more business roles that work together to perform collective behavior.</td>
<td><img src="image" alt="Business collaboration" /></td>
</tr>
<tr>
<td>Business interface</td>
<td>A point of access where a business service is made available to the environment.</td>
<td><img src="image" alt="Business interface" /></td>
</tr>
<tr>
<td>Location</td>
<td>A conceptual point or extent in space.</td>
<td><img src="image" alt="Location" /></td>
</tr>
<tr>
<td>Business process</td>
<td>A behavior element that groups behavior based on an ordering of activities. It is intended to produce a defined set of products or business services.</td>
<td><img src="image" alt="Business process" /></td>
</tr>
<tr>
<td>Business function</td>
<td>A behavior element that groups behavior based on a chosen set of criteria (typically required business resources and/or competences).</td>
<td><img src="image" alt="Business function" /></td>
</tr>
<tr>
<td>Business interaction</td>
<td>A behavior element that describes the behavior of a business collaboration.</td>
<td><img src="image" alt="Business interaction" /></td>
</tr>
</tbody>
</table>
## Business Layer Concepts (2/2)

<table>
<thead>
<tr>
<th>Concept</th>
<th>Description</th>
<th>Notation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business event</td>
<td>Something that happens (internally or externally) and influences behavior.</td>
<td><img src="image" alt="Business event" /></td>
</tr>
<tr>
<td>Business service</td>
<td>A service that fulfills a business need for a customer (internal or external to the organization).</td>
<td><img src="image" alt="Business service" /></td>
</tr>
<tr>
<td>Business object</td>
<td>A passive element that has relevance from a business perspective.</td>
<td><img src="image" alt="Business object" /></td>
</tr>
<tr>
<td>Representation</td>
<td>A perceptible form of the information carried by a business object.</td>
<td><img src="image" alt="Representation" /></td>
</tr>
<tr>
<td>Meaning</td>
<td>The knowledge or expertise present in a business object or its representation, given a particular context.</td>
<td><img src="image" alt="Meaning" /></td>
</tr>
<tr>
<td>Value</td>
<td>The relative worth, utility, or importance of a business service or product.</td>
<td><img src="image" alt="Value" /></td>
</tr>
<tr>
<td>Product</td>
<td>A coherent collection of services, accompanied by a contract/set of agreements, which is offered as a whole to (internal or external) customers.</td>
<td><img src="image" alt="Product" /></td>
</tr>
<tr>
<td>Contract</td>
<td>A formal or informal specification of agreement that specifies the rights and obligations associated with a product.</td>
<td><img src="image" alt="Contract" /></td>
</tr>
</tbody>
</table>
Business Layer Meta-model
Example
MOTIVATION Extension

Motivation Modeling provides the context or reason lying behind the architecture of a solution design.
## Business Motivation Concepts

<table>
<thead>
<tr>
<th>Concept</th>
<th>Definition</th>
<th>Notation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stakeholder</td>
<td>The role of an individual, team, or organization (or classes thereof) that represents their interests in, or concerns relative to, the outcome of the architecture.</td>
<td><img src="image" alt="Stakeholder" /></td>
</tr>
<tr>
<td>Driver</td>
<td>Something that creates, motivates, and fuels the change in an organization.</td>
<td><img src="image" alt="Driver" /></td>
</tr>
<tr>
<td>Assessment</td>
<td>The outcome of some analysis of some driver.</td>
<td><img src="image" alt="Assessment" /></td>
</tr>
<tr>
<td>Goal</td>
<td>An end state that a stakeholder intends to achieve.</td>
<td><img src="image" alt="Goal" /></td>
</tr>
<tr>
<td>Requirement</td>
<td>A statement of need that must be realized by a system.</td>
<td><img src="image" alt="Requirement" /></td>
</tr>
<tr>
<td>Constraint</td>
<td>A restriction on the way in which a system is realized.</td>
<td><img src="image" alt="Constraint" /></td>
</tr>
<tr>
<td>Principle</td>
<td>A normative property of all systems in a given context, or the way in which they are realized.</td>
<td><img src="image" alt="Principle" /></td>
</tr>
</tbody>
</table>
Business Motivation Meta-model
Example
IMPLEMENTATION / MIGRATION Extension

Implementation/Integration Meta-model

<table>
<thead>
<tr>
<th>Concept</th>
<th>Definition</th>
<th>Notation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Work Package</td>
<td>A series of actions designed to accomplish a major goal within a specified time</td>
<td>Work package</td>
</tr>
<tr>
<td>Deliverable</td>
<td>A precisely-defined outcome of a work package</td>
<td>Deliverable</td>
</tr>
<tr>
<td>Plateau</td>
<td>A relatively stable state of the architecture that exists during a limited period of time</td>
<td>Plateau</td>
</tr>
<tr>
<td>Gap</td>
<td>An outcome of a gap analysis between two plateaus</td>
<td>Gap</td>
</tr>
</tbody>
</table>
# Proposed Concepts

<table>
<thead>
<tr>
<th>Concept</th>
<th>Definition</th>
<th>Notation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Work Package</td>
<td>A series of actions designed to accomplish a unique goal within a specified time.</td>
<td><img src="image" alt="Work package" /></td>
</tr>
<tr>
<td>Deliverable</td>
<td>A precisely-defined outcome of a work package.</td>
<td><img src="image" alt="Deliverable" /></td>
</tr>
<tr>
<td>Plateau</td>
<td>A relatively stable state of the architecture that exists during a limited period of time.</td>
<td><img src="image" alt="Plateau" /></td>
</tr>
<tr>
<td>Gap</td>
<td>An outcome of a gap analysis between two plateaus.</td>
<td><img src="image" alt="Gap" /></td>
</tr>
</tbody>
</table>
Implementation/Integration Meta-model
Example

Baseline architecture

Gap baseline-target

Plateau

New hardware configuration

Integrated back-office suite

Hardware update

Software modification

Legacy outphasing

Back-office system integration project

Target architecture

© The Open Group
Mapping Archimate to Rows 3/4/5
The Application layer supports the business layer with application services which are realised by (software) applications.
## Application/Data Layer Concepts

<table>
<thead>
<tr>
<th>Concept</th>
<th>Definition</th>
<th>Notation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application component</td>
<td>A modular, deployable, and replaceable part of a software system that encapsulates its behavior and data and exposes these through a set of interfaces.</td>
<td><img src="attachment" alt="Application component" /></td>
</tr>
<tr>
<td>Application collaboration</td>
<td>An aggregate of two or more application components that work together to perform collective behavior.</td>
<td><img src="attachment" alt="Application collaboration" /></td>
</tr>
<tr>
<td>Application interface</td>
<td>A point of access where an application service is made available to a user or another application component.</td>
<td><img src="attachment" alt="Application interface" /></td>
</tr>
<tr>
<td>Application function</td>
<td>A behavior element that groups automated behavior that can be performed by an application component.</td>
<td><img src="attachment" alt="Application function" /></td>
</tr>
<tr>
<td>Application interaction</td>
<td>A behavior element that describes the behavior of an application collaboration.</td>
<td><img src="attachment" alt="Application interaction" /></td>
</tr>
<tr>
<td>Application service</td>
<td>A service that exposes automated behavior.</td>
<td><img src="attachment" alt="Application service" /></td>
</tr>
<tr>
<td>Data object</td>
<td>A passive element suitable for automated processing.</td>
<td><img src="attachment" alt="Data object" /></td>
</tr>
</tbody>
</table>
Application Layer Meta-model

Constraint:
Application Interaction may only be assigned by an Application Collaboration, not by an Application Component.
Example
Mapping Archimate to Rows 5/6
The Technology layer offers infrastructure services (e.g., processing, storage and communication services) needed to run applications, realised by computer and communication hardware and system software.
# Technology/Infrastructure Layer Concepts

<table>
<thead>
<tr>
<th>Concept</th>
<th>Definition</th>
<th>Notation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node</td>
<td>A computational resource upon which artifacts may be stored or deployed for execution.</td>
<td><img src="image" alt="Node" /></td>
</tr>
<tr>
<td>Device</td>
<td>A hardware resource upon which artifacts may be stored or deployed for execution.</td>
<td><img src="image" alt="Device" /></td>
</tr>
<tr>
<td>Network</td>
<td>A communication medium between two or more devices.</td>
<td><img src="image" alt="Network" /></td>
</tr>
<tr>
<td>Communication path</td>
<td>A link between two or more nodes, through which these nodes can exchange data.</td>
<td><img src="image" alt="Communication path" /></td>
</tr>
<tr>
<td>Infrastructure interface</td>
<td>A point of access where infrastructure services offered by a node can be accessed by other nodes and application components.</td>
<td><img src="image" alt="Infrastructure interface" /></td>
</tr>
<tr>
<td>System software</td>
<td>A software environment for specific types of components and objects that are deployed on it in the form of artifacts.</td>
<td><img src="image" alt="System software" /></td>
</tr>
<tr>
<td>Infrastructure function</td>
<td>A behavior element that groups infrastructural behavior that can be performed by a node.</td>
<td><img src="image" alt="Infrastructure function" /></td>
</tr>
<tr>
<td>Infrastructure service</td>
<td>An externally visible unit of functionality, provided by one or more nodes, exposed through well-defined interfaces, and meaningful to the environment.</td>
<td><img src="image" alt="Infrastructure service" /></td>
</tr>
<tr>
<td>Artifact</td>
<td>A physical piece of data that is used or produced in a software development process, or by deployment and operation of a system.</td>
<td><img src="image" alt="Artifact" /></td>
</tr>
</tbody>
</table>
Technology Layer Meta-model
Example
Archimate Viewpoints
...i.e. Formalized, Verifiable
Review of a few Viewpoints
Review of a few Viewpoints

Archimate® 2.1

Archimate Standard Viewpoints

<table>
<thead>
<tr>
<th>Viewpoint</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introductory</td>
<td>This viewpoint uses a simplified notation to explain the essence of an architecture model to non-architects that require a simpler, more intuitive notation.</td>
</tr>
<tr>
<td>Organization</td>
<td>This viewpoint focuses on the (internal) organization of a company, a department, a network of companies, or of another organizational entity.</td>
</tr>
<tr>
<td>Actor Co-operation</td>
<td>This viewpoint focuses on the relationships of actors with each other and their environment.</td>
</tr>
<tr>
<td>Business Function</td>
<td>This viewpoint shows the main business functions of an organization and their relationships in terms of the flows of information, value, or goods between them.</td>
</tr>
<tr>
<td>Business Process</td>
<td>This viewpoint shows the high-level structure and composition of one or more business processes.</td>
</tr>
<tr>
<td>Business Process Co-operation</td>
<td>This viewpoint shows the relationships of one or more business processes with each other and/or with their environment.</td>
</tr>
<tr>
<td>Product</td>
<td>This viewpoint describes the value that one or more products offer to the customers or other external parties involved and shows the composition of one or more products in terms of the constituting (business or application) services, and the associated contract(s) or other agreements.</td>
</tr>
<tr>
<td>Application Behavior</td>
<td>This viewpoint describes the internal behavior of an application; e.g., as it realizes one or more application services.</td>
</tr>
<tr>
<td>Application Co-operation</td>
<td>This viewpoint describes the relationships between applications components in terms of the information flows between them, or in terms of the services they offer and use.</td>
</tr>
<tr>
<td>Application Structure</td>
<td>This viewpoint shows the structure of one or more applications or components.</td>
</tr>
<tr>
<td>Application Usage</td>
<td>This viewpoint describes how applications are used to support one or more business processes, and how they are used by other applications.</td>
</tr>
<tr>
<td>Infrastructure</td>
<td>This viewpoint describes the software and hardware infrastructure elements supporting the application layer, such as physical devices, networks, or system software (e.g., operating systems, databases, and middleware).</td>
</tr>
<tr>
<td>Infrastructure Usage</td>
<td>This viewpoint shows how applications are supported by the software and hardware infrastructure: the infrastructure services are delivered by the devices, system software and networks are provided to the applications.</td>
</tr>
<tr>
<td>Implementation and Deployment</td>
<td>This viewpoint shows how one or more applications are realized on the infrastructure.</td>
</tr>
<tr>
<td>Information Structure</td>
<td>This viewpoint shows the structure of the information used in the enterprise or in a specific business process or application, in terms of data types or (object-oriented) class structures.</td>
</tr>
<tr>
<td>Service Realization</td>
<td>This viewpoint shows how one or more business services are realized by the underlying processes (and sometimes by application components).</td>
</tr>
<tr>
<td>Layered</td>
<td>This viewpoint shows several layers and aspects of an enterprise architecture in a single diagram.</td>
</tr>
<tr>
<td>Landscape Map</td>
<td>This viewpoint uses a matrix to represent a three-dimensional co-ordinate system describing architectural relationships.</td>
</tr>
</tbody>
</table>

N132 Reference Card: Archimate® 2.1 Viewpoints

Copyright © 2013 The Open Group. All Rights Reserved.
Archimate® is a registered trademark of The Open Group.
Review of a few Viewpoints

For the complete list / description of Archimate Viewpoints, refer to the Archimate 2.1 Specification included in Moodle, or review it online at [http://pubs.opengroup.org/architecture/archimate2-doc/].
Example: Actor Co-operation Viewpoint

- To determine collaborations in the value chain or network in which actors co-operate.
- To outline relations of actors (roles, teams, departments) with each other and their environment (third-parties, customers, intermediaries).
- To determine internal and external dependencies.
- Model-type(s): Context diagrams or Value Chain diagrams.
Example: Application Behavior Viewpoint
Example: Application Behavior Viewpoint

To describe the internal behaviour of an application; e.g., as it realizes one or more application services.

To design the main behaviour of applications.

...helpful in identifying functional overlap between different applications (i.e. cross cutting concern, impact analysis)

.. or identifying functional coverage gaps within or between applications.

Linked Model-type(s): Behavioral, Data Flow diagram.
Meta-model
Example: Application Co-operation Viewpoint
Example: Application Co-operation Viewpoint

To express the (internal) orchestration of application services that together support the execution of a business process.
To describe information flowing between application components.
To enumerate the services application components offer or/and use.
Can be used to create a high-level overview of the application landscape of an organization.
Linked Model-type(s): Behavioral, Data Flow or Component collaboration diagrams.
Meta-model
Example: Application Structure Viewpoint
Example: Application Structure Viewpoint

To outline the structure of one or more applications or components.
To associate components and data entities.
Linked Model-type(s): Functional, Component/Application diagrams.
Meta-model
Example: Business Co-operation Viewpoint

Example

Meta-model
Example: Business Co-operation Viewpoint

To outline causal relations of business processes with each other and/or with their environment.
To outline business process dependencies within their temporal context.
To show how what business process realize what Business services.
To outline cross-cutting uses of shared data.
To show actors accountable for / interacting with process steps.
Model-kind: Behavioral, Process Flow diagrams.
Meta-model
Example: Business Function Viewpoint

• To show domains of capabilities/functions of a solution.
• To outline primary activities, technological features.
• To define relation to departments/teams/roles and define accountability/ownership.

Model-type: Capability Map diagrams, Bounded-context diagrams.
Meta-model
Example: Business Process Viewpoint
Example: Business Process Viewpoint

To show the high-level structure of one or more business processes. To show information types required/used for each process step. To show actors interacting with each process step and define accountability. To realize service offered / exposed. To compose/relate processes with other processes. Model-type: Behavioral
Meta-model
Example: Business Product Viewpoint

Meta-model
Example: Business Product Viewpoint

To depict the value products offer to external parties (intermediaries, customers,..)

To show the composition of one or more products in terms of the constituting (business or application) services, and the associated contract(s) or other agreements.

To show the interfaces (channels) through which this product is offered, and the events associated with the product.

To identify which services can be re-used, or must be created for a product, given the value a customer expects from it.

Product are realized by processes.

Model-type: Structural, Service Map
Meta-model
Example: Information Structure Viewpoint
Example: Information Structure Viewpoint

To show the structure of the information used in the solution.
To map data to business processes or application function/services, in terms of data types/messages or data contract structures.
To realize business information concepts with data structures used.
To allocate data assets to underlying infrastructure; e.g., by means of a database schema.
Model-kind: Data Structure
Meta-model
Example: Organization Viewpoint
Example: Organization Viewpoint

To model the (internal) organisation of a company, a department, a network of companies, or of another organisational entity. To identifying geographies, and map competencies, activities. Model-kind: Organizational Chart, Nested-block diagrams.
Meta-model
Example: Service Realization Viewpoint
Example: Service Realization Viewpoint

To show how one or more business services are realized by the underlying processes (and sometimes by application components).
To realize business process flows with application services.
Model-kind: Function <-> Behavioral Mappings
Meta-model
Example: Infrastructure Viewpoint

Meta-model
Example: Infrastructure Viewpoint

The Infrastructure viewpoint contains the software and hardware infrastructure elements supporting the application layer, such as physical devices, networks, or system software (e.g., operating systems, databases, and middleware).
Meta-model

Diagram showing relationships between artifacts, infrastructure services, infrastructure function, node, communication path, location, device, and network.
Example: Layered Viewpoint
Example
# Extended Viewpoints

## ArchiMate® 2.1

### ArchiMate Extension Viewpoints

<table>
<thead>
<tr>
<th>Viewpoint</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stakeholder</td>
<td>Motivation Extension</td>
<td>This viewpoint allows the analyst to model the stakeholders, the internal and external drivers for change, and the assessments (in terms of strengths, weaknesses, opportunities, and threats) of these drivers.</td>
</tr>
<tr>
<td>Goal Realization</td>
<td>Motivation Extension</td>
<td>This viewpoint allows a designer to model the refinement of (high-level) goals into more concrete goals, and the refinement of concrete goals into requirements or constraints that describe the properties that are needed to realize the goals.</td>
</tr>
<tr>
<td>Goal Contribution</td>
<td>Motivation Extension</td>
<td>This viewpoint allows a designer or analyst to model the influence relationships between goals and requirements.</td>
</tr>
<tr>
<td>Principles</td>
<td>Motivation Extension</td>
<td>This viewpoint allows the analyst or designer to model the principles that are relevant to the design problem at hand, including the goals that motivate these principles.</td>
</tr>
<tr>
<td>Requirements Realization</td>
<td>Motivation Extension</td>
<td>This viewpoint allows the designer to model the realization of requirements by the core elements, such as business actors, business services, business processes, application services, application components, etc.</td>
</tr>
<tr>
<td>Motivation</td>
<td>Motivation Extension</td>
<td>This viewpoint allows the designer or analyst to model the motivation aspect, without focusing on certain elements within this aspect.</td>
</tr>
<tr>
<td>Project</td>
<td>Implementation &amp; Migration Extension</td>
<td>This viewpoint is used to model the management of architecture change.</td>
</tr>
<tr>
<td>Migration</td>
<td>Implementation &amp; Migration Extension</td>
<td>This viewpoint contains models and concepts that describe the transition from an existing architecture to a desired architecture.</td>
</tr>
<tr>
<td>Implementation &amp; Migration</td>
<td>Implementation &amp; Migration Extension</td>
<td>This viewpoint is used to relate programs and projects to the parts of the architecture that they implement.</td>
</tr>
</tbody>
</table>
Meta-model
Example