INF5120 and INF9120
”Modelbased System development”

Lecture 6: 18.02.2018
Arne-Jørgen Berre

arneb@ifi.uio.no and Arne.J.Berre@sintef.no
### Course parts – 2019
(tentative plan)

<table>
<thead>
<tr>
<th>Part I-a -(Introduction to Modeling):</th>
</tr>
</thead>
<tbody>
<tr>
<td>14/1 - Start</td>
</tr>
<tr>
<td>1-14/1: Introduction to INF5120</td>
</tr>
<tr>
<td>Part I-b -(Introduction to Modeling and Metamodeling and)</td>
</tr>
<tr>
<td>2-21/1: Modeling structure and behaviour (UML and UML 2.0 and metamodeling)</td>
</tr>
<tr>
<td>3-28/1: MDE with Eclipse EMF, Xtext and SIRIUS</td>
</tr>
<tr>
<td>4-4/2: Business Modeling – BMC, LSC, VDML/VDMBee, (w/meta model) - (Business Model Canvas, Value Proposition, Lean Canvas - VDMBee, VDML</td>
</tr>
<tr>
<td>Part II - (Business Architecture/Business Modeling Modeling + Server side – Node-RED + THingML + Sensors/IoT/Cloud - and the Smart Home project):</td>
</tr>
<tr>
<td>5-11/2: BPMN</td>
</tr>
<tr>
<td>6-18/2: ArchiMate 3.0 Modeling, TOGAF 9.0</td>
</tr>
<tr>
<td>7-25/2: Node-RED Cognitive Services/AI – Watson services example (for Oblig 2) (IBM Cloud)</td>
</tr>
<tr>
<td>8-4/3: Smart Home - Design Thinking (with IBM)</td>
</tr>
<tr>
<td>4/3 - Oblig 1: Individual / Pairs - Metamodeling + Node-RED (pass / no-pass)</td>
</tr>
<tr>
<td>10- 18/3: UML 2.x Sequence Diagrams, UML 2.x State Machines. ArchiMate ArchInsurance, ArhiMetal examples,</td>
</tr>
<tr>
<td>11- 25/3 SysML and Arcadia/Capella and CJML</td>
</tr>
<tr>
<td>25/3 - Oblig 2: Specification Models Smart Home – Business Architecture – Business Canvases, ++, ArchiMate + IBM</td>
</tr>
<tr>
<td>Part IIi - (Modeling –Apps + EA/ArchiMate)</td>
</tr>
<tr>
<td>11- 1/4 UI – Server, UI Start lecture – incl. IFML – EA with ArchiMate - Oblig 2 (3) discussion/presentation</td>
</tr>
<tr>
<td>Part IV - (MDE – Creating Your own Domain Specific Language – Eclipse, Xtext/SIRIUS):</td>
</tr>
<tr>
<td>12-8/4: Model driven engineering – revisited</td>
</tr>
<tr>
<td>Oblig 3: Smart Home – Realisation with Node-RED, IBM Watson/IoT services + ArchiMate/UML++ Application, Technology, Physical models – 10%</td>
</tr>
<tr>
<td>EASTER – 15/4 and 22/4</td>
</tr>
<tr>
<td>29/4 - Oblig 3 - delivery</td>
</tr>
<tr>
<td>13-29/4: MDE – Lexical (ThingML) and Graphical (CJML) DSLs and Transformations</td>
</tr>
<tr>
<td>14-6/5 Lecture – Model Driven Engineering</td>
</tr>
<tr>
<td>15-13/5 Lecture - Model Driven Engineering</td>
</tr>
<tr>
<td>13/5 – Oblig 4- Smart Home demonstrations - 10%</td>
</tr>
<tr>
<td>16-20/5: Summary of the course - Oblig 4 final discussions - Conclusions, Previous exams – group collaborations</td>
</tr>
<tr>
<td>Preparations for the Exam by old exams</td>
</tr>
</tbody>
</table>

- 3/6: Exam (4 hours), Mandag June 3rd, (0900-1300) - 70%
This lecture, February 18, 2019

ArchiMate 3.0 + TOGAF 9.0

+ Design Thinking (prepare for Oblig 2)

- Enterprise Architecture
- TOGAF
- ArchiMate
- Modelio Archimate, BPMN and UML modelling tool
- ArchiMate Business Process
Content

- Enterprise Architecture
- TOGAF
- ArchiMate og Archi and Modelio
<table>
<thead>
<tr>
<th>Perspective</th>
<th>Audit</th>
<th>What</th>
<th>How</th>
<th>Where</th>
<th>Align</th>
<th>Who</th>
<th>Why</th>
</tr>
</thead>
<tbody>
<tr>
<td>Executive</td>
<td></td>
<td>Inventory Identification</td>
<td>Process Identification</td>
<td>Distribution Identification</td>
<td>Responsibility Identification</td>
<td>Timing Identification</td>
<td>Motivation Identification</td>
</tr>
<tr>
<td>Business Mgmt</td>
<td></td>
<td>Inventory Definition</td>
<td>Process Definition</td>
<td>Distribution Definition</td>
<td>Responsibility Definition</td>
<td>Timing Definition</td>
<td>Motivation Definition</td>
</tr>
<tr>
<td>Architecture</td>
<td></td>
<td>Inventory Representation</td>
<td>Process Representation</td>
<td>Distribution Representation</td>
<td>Responsibility Representation</td>
<td>Timing Representation</td>
<td>Motivation Representation</td>
</tr>
<tr>
<td>Technician</td>
<td></td>
<td>Inventory Configuration</td>
<td>Process Configuration</td>
<td>Distribution Configuration</td>
<td>Responsibility Configuration</td>
<td>Timing Configuration</td>
<td>Motivation Configuration</td>
</tr>
<tr>
<td>Enterprise</td>
<td></td>
<td>Inventory Instantiations</td>
<td>Process Instantiations</td>
<td>Distribution Instantiations</td>
<td>Responsibility Instantiations</td>
<td>Timing Instantiations</td>
<td>Motivation Instantiations</td>
</tr>
</tbody>
</table>

**Composite Integrations**

**Alignment**

**Telecom and Informatics**

5
Zachman Framework

- **Row 1 – Scope**
  External Requirements and Drivers
  Business Function Modeling

- **Row 2 – Enterprise Model**
  Business Process Models

- **Row 3 – System Model**
  Logical Models
  Requirements Definition

- **Row 4 – Technology Model**
  Physical Models
  Solution Definition and Development

- **Row 5 – As Built**
  As Built
  Deployment

- **Row 6 – Functioning Enterprise**
  Functioning Enterprise
  Evaluation
Many Architectural Frameworks ....

ARIS

ZACHMAN

GERAM

EN/ISO 19439

Athena OEA

NIST
TOGAF 9 (The Open Group)
The TOGAF Architecture Development Method (ADM)
TOGAF 9 (The Open Group)
Open Group ADM

Diagram:
- Preliminary
- A. Architecture Vision
- B. Business Architecture
- C. Information Systems Architectures
- D. Technology Architecture
- E. Opportunities and Solutions
- F. Migration Planning
- G. Implementation Governance
- H. Architecture Change Management

Diagram structure:
- Requirements Management

Note: The diagram shows a flow of dependencies between the components.
### Preliminary
- Catalogs
- Principles Catalog

### Architecture Vision
- Core Diagrams
  - Value Chain Diagram
  - Solution Concept Diagram

### Business Architecture
- Catalogs
  - Organization / Actor Catalog
  - Driver / Goal / Objective Catalog
  - Role Catalog
  - Service / Function Catalog
  - Location Catalog
  - Process / Event / Control / Product Catalog
  - Contract / Measure Catalog

- Matrices
  - Business Interaction Matrix
  - Actor / Role Matrix

- Core Diagrams
  - Business Footprint Diagram
  - Service / Information Diagram
  - Functional Decomposition Diagram
  - Product Lifecycle Diagram

- Extension Diagrams
  - Goal/Objective/Service Diagram
  - Business Use-Case Diagram
  - Organization Decomposition Diagram
  - Process Flow Diagram
  - Event Diagram

### Data Architecture
- Catalogs
  - Data Entity / Data Component Catalog

- Matrices
  - Data Entity / Business Function Matrix
  - System / Data Matrix

- Core Diagrams
  - Class Diagram
  - Data Dissemination Diagram

- Extension Diagrams
  - Data Security Diagram
  - Class Hierarchy Diagram
  - Data Migration Diagram
  - Data Lifecycle Diagram

### Application Architecture
- Catalogs
  - Application Portfolio Catalog

- Matrices
  - System / Organization Matrix
  - Role / System Matrix
  - System / Function Matrix
  - Application Interaction Matrix

- Core Diagrams
  - Application Communication Diagram
  - Application and User Location Diagram
  - System Use-Case Diagram

- Extension Diagrams
  - Enterprise Manageability Diagram
  - Process / System Realization Diagram
  - Software Engineering Diagram
  - Application Migration Diagram
  - Software Distribution Diagram

### Technology Architecture
- Catalogs
  - Technology Standards Catalog
  - Technology Portfolio Catalog

- Matrices
  - System / Technology Matrix

- Core Diagrams
  - Environments and Locations Diagram
  - Platform Decomposition Diagram

- Extension Diagrams
  - Processing Diagram
  - Networked Computing / Hardware Diagram
  - Communications Engineering Diagram

### Requirements Management
- Catalogs
  - Requirements Catalog

### Opportunities and Solutions
- Core Diagrams
  - Project Context Diagram

- Benefits Diagram
Building block evolution

A. Architecture Vision
   - High-level model of candidate building blocks

B. Business Architecture
C. Data/Application Architecture
D. Technology Architecture

Step 1: Select Reference Models, Viewpoints, and Tools
Step 2: Develop Baseline Architecture Description
   - High-level model of existing building blocks, re-using definitions from the Architecture Repository where they are available

Step 3: Develop Target Architecture Description
   - Develop view of required building blocks through the creation of catalogs, matrices, and diagrams of the architecture
   - Fully document each building block
   - Document rationale for building block decisions in architecture document
   - Identify the impacted building blocks, checking against a library of building blocks within the Architecture Repository and re-using where appropriate
   - Where necessary, define new building blocks
   - Select standards for each building block, re-using as much as possible from reference models selected from the Architecture Continuum
   - Document final mapping of the building blocks to the Architecture Landscape
   - From selected building blocks, identify those that might be re-used, and publish as standards or reference models via the Architecture Repository

Step 4: Perform Gap Analysis
   - Identify building blocks carried over
   - Identify eliminated building blocks
   - Identify new building blocks
   - Identify gaps and determine realization approach (e.g., to be developed or to be procured)

Step 5: Define Roadmap Components
Step 6: Resolve Impacts across the Architecture Landscape
Step 7: Formal Stakeholder Review
Step 8: Finalize the Architecture
Step 9: Create the Architecture Definition Document
Service categories

- Infrastructure Applications
- Business Applications
- Application Platform Interface
- System and Network Management
- Software Engineering
- Security
- Transaction Processing
- Location & Directory
- International Operations
- User Interface
- Data Interchange
- Data Management
- Graphics & Image
- Operating System Services
- Network Services
- Communications Infrastructure Interface
- Communications Infrastructure
TOGAF

TOGAF is a framework for EA which provides a comprehensive approach to the design, planning, implementation, and governance of an enterprise information architecture. TOGAF is a registered trademark of The Open Group. TOGAF is a high level and holistic approach to design, which is typically modeled at four levels: Business, Application, Data, and Technology. As an open standard, TOGAF is widely adopted to support EA.

Togaf Modeling

"Togaf Modeling" is a collaborative site designed to provide a guide and practical modeling solutions, by selecting different standards that provide an off-the-shelf solution. There is no universal solution, but our aim is to identify a practical solution, through your contributions.

http://www.togaf-modeling.org/
A solution concept diagram provides a high-level orientation of the solution that is envisaged in order to meet the objectives of the architecture engagement. In contrast to the more formal and detailed architecture diagrams developed in the following phases, the solution concept represents a pencil sketch of the expected solution at the outset of the engagement. This diagram may embody key objectives, requirements and constraints for the engagement, and also highlight work areas to be investigated in more detail with formal architecture modeling. The purpose of this diagram is to quickly on-board and align stakeholders for a particular change initiative, so that all participants understand what the architecture engagement is seeking to achieve and how it is expected that a particular solution approach will meet the needs of the enterprise.
Archimate 3.0

- http://www.opengroup.org/subjectareas/enterprise/archimate
Archimate 3.0

- What is new:

- Introduction to Archimate 3.0 – part 1 to 7 (vode=
  - https://www.youtube.com/watch?v=ULI9lf0OZco&list=PLB8F2ECDADCEE616AA

- Archimate 3.0 in Practice – part 1 to 5
  - https://www.youtube.com/watch?v=_kmYkxKb-o&list=PLB8F2ECDADCEE616AA&index=8
About ArchiMate

ArchiMate®, an Open Group Standard, is an open and independent modelling language for enterprise architecture that is supported by different tool vendors and consulting firms. ArchiMate provides instruments to enable enterprise architects to describe, analyze and visualize the relationships among business domains in an unambiguous way.

Just as an architectural drawing in classical building architecture describes the various aspects of the construction and use of a building, ArchiMate offers a common language for describing the construction and operation of business processes, organizational structures, information flows, IT systems, and technical infrastructure. This insight helps stakeholders to design, assess, and communicate the consequences of decisions and changes within and between these business domains.

The ArchiMate 2.1 Specification

The ArchiMate 2.1 Specification is a maintenance update to ArchiMate 2.0, addressing comments raised since the introduction of ArchiMate 2.0 in 2012. The ArchiMate Specification is based on many years of practical experience of modeling and analysis of Enterprise Architecture (EA) by a world-wide user base. It enables the creation of fully integrated models of the organization’s enterprise architecture, the motivation for it, and the programs, projects and migration paths to implement it.

The detailed changes between version 2.0 and 2.1 are available as Document U132: ArchiMate 2.0 Specification Technical Corrigendum Number 1

The ArchiMate modeling language has evolved to be fully aligned with the TOGAF® standard:

- By providing a vendor-independent set of concepts, that helps to create a consistent, integrated model “below the waterline”, which can be depicted in the form of TOGAF views
- The ArchiMate® language enables modeling throughout the TOGAF® Architecture Development Method
- The language structure of the ArchiMate Core corresponds with the three main architectures as addressed in phases B, C & D in the TOGAF® ADM
- The extensions to the Core closely correspond with the main aspects to be addressed in the Preliminary phase, Phase A and the Central Requirements management repository, as well as Phases E, F, G and H
Archi

http://www.archimatetool.com/

The Free ArchiMate Modelling Tool

A free and open source modelling tool to create ArchiMate models and sketches. Used by hundreds of Enterprise Architects throughout the world.
MODELIO – to be used in the course!

Welcome to Modelio
The Open Source Modeling Environment

Add ArchiMate 3.0 support for free
Download and install the Modelio ArchiMate plugin and start modeling with the ArchiMate standard!

Getting started
Quick tour of Modelio features... Worth a try!

UML practical guides
A set of practical and pragmatic UML modeling guides...

Migrating projects
How to migrate earlier Modelio projects...
Modeliosoft is proud to introduce our new Modelio 3.6 version, which still increases its benefits by bringing new exclusive features.

**New Modelio BA ArchiMate EA**

Modelio BA ArchiMate EA is the only tool on the market, supporting natively ArchiMate, UML and BPMN. This provides a modeling capacity that completely covers Enterprise modeling needs, and provide a complete and consistent repository throughout the entire enterprise.

- ArchiMate, UML and BPMN Modeling, and mapping management
- Integrated Vision and requirement analysis support
- Documentation Generation
- Spreadsheets & matrices editors
- Traceability and impact analysis management
- Integration of model processing code (Java, Jython, ...)

---

**Telecom and Informatics**

---

**SINTEF**
Archimate for Modelio 3.6.x

Documents

Archimate - Quick Start Guide (en) - hot

This quick tutorial will guide you through the process of creating your first ArchiMate model in Modelio

Date added: 11/08/2016

« Start Prev 1 2 Next End »

Page 2 of 2
ArchiMate Downloads

How do I download the ArchiMate 3.0 Specification?

- Downloading ArchiMate 3.0 as an ArchiMate Forum member
- Downloading ArchiMate 3.0 as a non-member

Downloading the ArchiMate 3.0 Specification as an ArchiMate Forum Member

If your organization is a member of The Open Group ArchiMate Forum, you may download and use ArchiMate internally under a personal annual Member License.

The download procedure:

1. Check the ArchiMate 3.0 Member Licensing Conditions; then
2. Register and download ArchiMate 3.0 under an Annual Member License.

Downloading ArchiMate 3.0 as a Non-Member

You may download ArchiMate Version 3.0 under a free, personal, 90-day Evaluation License, and you may then go on to use it internally under a free, perpetual Non-Commercial License. Alternatively, you may take out an annual Commercial License.

Applying for a ArchiMate 3.0 Evaluation License

The download procedure

1. Check the ArchiMate 3.0 Evaluation Licensing Conditions; then
2. Register and download ArchiMate 3.0 under a 90-day Evaluation License.

If you download ArchiMate, you are presumed to have read and accepted the ArchiMate 3.0 Evaluation Licensing Conditions. There is no need to complete and submit the license.

After the Evaluation Period

After expiry of the 90-day evaluation period you should apply for a more permanent License:

- A free, perpetual Non-Commercial License is available to Corporate, Academic Institutions and Individuals wanting to use the ArchiMate 3.0 Specification for non-commercial use.
- An annual Commercial License is available for organizations wishing to exploit ArchiMate 3.0 commercially.
UML, Use case diagram in Modelio
Business Product View

Customer

Be Insured

Travel Insurance

- Insurance Application Service
- Premium Payment Service
- Customer Information Service
- Claim Registration Service
- Customer data mutation
- Claims Payment Service

Travel Insurance Policy
ArchiMate viewpoint mechanism
Archimate

Authors: eSchoolink Group - ITNLU
Contents

1. What’s ArchiMate?
2. Why ArchiMate?
3. Main Benefits of ArchiMate
4. Layers of ArchiMate
5. ArchiMate vs UML
6. Notations of ArchiMate
7. Demo
What is ArchiMate?

- ArchiMate is a modelling technique ("language") for describing enterprise architectures.
- It presents a clear set of concepts within and relationships between architecture domains, and offers a simple and uniform structure for describing the contents of these domains.
- ArchiMate distinguishes itself from other languages such as Unified Modeling Language (UML) and Business Process Modeling Notation (BPMN) by its well defined metamodel, and wider enterprise modelling scope.
What is ArchiMate?

- ArchiMate offers a common language for describing the construction and operation of business processes, organizational structures, information flows, IT systems, and technical infrastructure.

- This insight helps the different stakeholders to design, assess, and communicate the consequences of decisions and changes within and between these business domains.
What is ArchiMate?

- An architecture framework is used to structure the concepts and relationships of the ArchiMate language.
- It divides the enterprise architecture into a business, application, and technology layer. In each layer, three aspects are considered: active elements that exhibit behavior (e.g., Process and Function), an internal structure, and elements that define use or communicate information.
Enterprise Architecture: Describing Coherence

Information architecture

Product architecture

Process architecture

Application architecture

Technical architecture

Telematica
Why ArchiMate?

- Enterprise architecture is an important instrument to address this company-wide integration.
- It is a coherent whole of principles, methods and models that are used in the design and realization of the enterprise's organizational structure, business processes, information systems, and IT infrastructure.
Why ArchiMate?

- A good architecture practice enables an organization to align business and IT operations with its strategy, quickly respond to changes in the environment, and make optimal use of technological opportunities.
- The development and maintenance of architectures will lead to efficiency, cost reduction and flexibility.
Why ArchiMate?

- Within companies various domain architectures can be found, like organization, business process, application, information, and technical architectures.
- Each architecture domain has its own concepts for the modelling and visualization of its internal coherence. These specific models and visualizations simplify communication, discussion and analysis within the domain.
Why ArchiMate?

- However, the relations between the concepts in these different domains are in many cases unclear. Moreover, these domains often partially overlap but use different notions to express the same ideas, sometimes even without the people involved knowing this.

- The resulting ambiguities and confusion stand in the way of the flexibly and efficiently operating organizations we envisage.
Why ArchiMate?

- ArchiMate wants to do away with these ambiguities. It presents a unified way of modelling enterprise architectures, integrating the various domains and describing them in an easily readable way.

- ArchiMate is of course not an isolated development. The relationships with existing methods and techniques, like modelling languages such as UML and BPMN, and methods and frameworks like TOGAF and Zachman, are well-described.
Main Benefits of ArchiMate

1. It is an international, vendor-independent standard of The Open Group, liberating you from the lock-in of vendor-specific tools and frameworks. There is active support from the ArchiMate Forum of The Open Group.

2. Its well-founded concepts and models provide precision. It helps you get away from the 'fuzzy pictures' image of architecture.

3. It is a lean and simple language. It contains just enough concepts for modeling enterprise architecture and is not bloated to include everything possible. Its uniform structure makes it easy to learn and apply.
Main Benefits of ArchiMate

4. It has clear links to **existing approaches** for specific architecture areas such as software or business processes. Several concepts in ArchiMate have deliberately been borrowed from other languages such as UML or BPMN, to provide an easy bridge.

5. It does not prescribe a way of working, but it is **easily combined** with existing methods such as TOGAF.

6. It has been **tried and tested** by many different user organizations and is **supported** by numerous consultancies and software tools.
A layered view provides a natural way to look at service-oriented models. The higher layers use services that are provided by the lower layers. ArchiMate distinguishes three main layers:

- The **Business layer** offers products and services to external customers, which are realized in the organization by business processes performed by business actors and roles.
- The **Application layer** supports the business layer with application services which are realized by (software) application components.
- The **Technology layer** offers infrastructural services (e.g., processing, storage and communication services) needed to run applications, realized by computer and communication hardware and system software.
Archimate vs UML

Archimate
- Archimate was created to model the architecture of an enterprise (all of the systems in an organization).
- Archimate models the business, information system (application and data), and technology architectures of the environment, including how these architectures are inter-related.

UML
- UML still functions best as a way to document the architecture of a single system
- UML provides 13 diagram types, providing flexibility to describe many different types of systems.
Archimate vs UML

Archimate started with an understanding that these problems relate to one another; that the entire complex and difficult business of understanding IT requires a rich inter-relationship of completely different domains, from business motivation to business process to managed services to systems to infrastructure.

Thus Archimate goes where UML doesn’t: it defines a metamodel that allows these relationships to be constructed, and constrained, and communicated. The constraints allow analysis, traceability, governance, and consistency. UML is unconstrained between model types. Archimate is not.
Notations

- Every concept and relation should have a precise graphical notation, with a sufficient resemblance the ‘standard’ ArchiMate notation. The notation in the Visio stencils can be used as a guideline.
- Optionally, multiple notations may exist for a single concept.
- It should be possible to denote composition, aggregation and assignment both with their ‘line’ notation and with nesting.
The following relation types should be supported:

- **Structural relations:**
  - composition*
  - aggregation
  - assignment
  - used by
  - realisation
  - access
  - association

- **Dynamic relations:**
  - triggering
  - flow

- **Other relations:**
  - grouping
  - junction
  - specialisation*
Notations
Demo

Product and Process

Travel Insurance
- Claim registration service
- Customer information service
- Claim payment service

Claim handling
- Registration
- Acceptance
- Valuation
- Payment

Policy
Notification
Letter
Demo

Actors, Roles, Processes

- Customer
- Insurant
- ArchiSurance
- Insurer

Claim registration service
Customer information service
Claim payment service

Claim handling
- Registration
- Acceptance
- Valuation
- Payment

Notification
Demo

Application Behavior

Policy creation service

Policy administration

Create policy

Assess risk → Calculate premium → Generate policy → Store policy

Insurance application

Insurance policy

Customer record
Demo

Application Usage

Claim handling

Registration → Acceptance → Valuation → Payment

Customer administration service

Claim administration service

Payment service

CRM system

Policy administration

Financial application

Notification

Notification data
Demo

Deployment

- CRM system
- Policy administration
- Financial application
- Notification data

- Database access service
- MQ messaging service
- Database tables

- zSeries Mainframe
  - DB2 Database

- Sun Blade
  - iPlanet App. server

- LAN

- Fin. application EJBs

Telematica
# ArchiMate 3.0 full framework

<table>
<thead>
<tr>
<th>Aspects</th>
<th>Passive structure</th>
<th>Behavior</th>
<th>Active structure</th>
<th>Motivation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strategy</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Business</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Application</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Technology</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Physical</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Implementation &amp; Migration</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

© 2016 The Open Group
Layers, domains

Business Layer
- Business actor
- Business role
- Business interface
- Business collaboration

Behavior
- Business service
- Business process
- Business function
- Business interaction
- Business event

Information
- Business object
- Representation
- Product
- Meaning
- Value
- Contract

Application Layer
- Archimate "Concepts"
  - Application component
  - Application interface
  - Application collaboration
  - Application service
  - Application function
  - Application interaction

Information
- Data object
- Artifact

Technology Layer
- Structure
  - Node
  - Device
  - Infrastructure interface
  - Network communication
  - Communication path
- Behavior
- Information

http://earchpal.wordpress.com/
Layers, domains

Business
- Business object
- Business process
- Business role
- Application service
- Application function
- Infrastructure service
- Artifact
- System software
- Device
- Network

Information
Behaviour
Structure
Overview of the ArchiMate concepts and main relationships.
<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
<th>Notation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business actor</td>
<td>A business entity that is capable of performing behavior.</td>
<td></td>
</tr>
<tr>
<td>Business role</td>
<td>The responsibility for performing specific behavior, to which an actor can be assigned, or the part an actor plays in a particular action or event.</td>
<td></td>
</tr>
<tr>
<td>Business collaboration</td>
<td>An aggregate of two or more business internal active structure elements that work together to perform collective behavior.</td>
<td></td>
</tr>
<tr>
<td>Business interface</td>
<td>A point of access where a business service is made available to the environment.</td>
<td></td>
</tr>
<tr>
<td>Business process</td>
<td>A sequence of business behaviors that achieves a specific outcome such as a defined set of products or business services.</td>
<td></td>
</tr>
<tr>
<td>Business function</td>
<td>A collection of business behavior based on a chosen set of criteria (typically required business resources and/or competences), closely aligned to an organization, but not necessarily explicitly governed by the organization.</td>
<td></td>
</tr>
<tr>
<td><strong>Business interaction</strong></td>
<td>A unit of collective business behavior performed by (a collaboration of) two or more business roles.</td>
<td></td>
</tr>
<tr>
<td>--------------------------</td>
<td>------------------------------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td><strong>Business event</strong></td>
<td>A business behavior element that denotes an organizational state change. It may originate from and be resolved inside or outside the organization.</td>
<td></td>
</tr>
<tr>
<td><strong>Business service</strong></td>
<td>An explicitly defined exposed business behavior.</td>
<td></td>
</tr>
<tr>
<td><strong>Business object</strong></td>
<td>A concept used within a particular business domain.</td>
<td></td>
</tr>
<tr>
<td><strong>Contract</strong></td>
<td>A formal or informal specification of an agreement between a provider and a consumer that specifies the rights and obligations associated with a product and establishes functional and non-functional parameters for interaction.</td>
<td></td>
</tr>
<tr>
<td><strong>Representation</strong></td>
<td>A perceptible form of the information carried by a business object.</td>
<td></td>
</tr>
<tr>
<td><strong>Product</strong></td>
<td>A coherent collection of services and/or passive structure elements, accompanied by a contract/set of agreements, which is offered as a whole to (internal or external) customers.</td>
<td></td>
</tr>
</tbody>
</table>
Overview of the ArchiMate concepts and main relationships.
ArchiMate - Behavior and Structure Elements Metamodel
Motivation aspect metamodel
Motivation and core elements
Example of motivation aspects

- Improve Profitability of Service Offering
- Increased Profit
- Increased Revenue
- Increased Market Share
- Reduced Cost of Customer Acquisition
- Serve Customers Wherever They Are
- Serve Customers Whenever They Need Our Help
- Respond To Changing Customer Needs, Preferences, And Expectations Quickly And Efficiently
- Mobile Applications Shall Run On All Popular Mobile Platforms
- Services Shall Be Accessible Through Mobile Browsers
- Mobile Applications Shall Be Built With Cross-Platform Frameworks
Strategy layer metamodel
Example Strategy Model
Relationships between Strategy Elements and Motivation and Core Elements
UML USE CASES AND USER STORIES
Use case modeling
## Template of a Use Case Description

<table>
<thead>
<tr>
<th>Use Case Template</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use Case Name</td>
<td>Visualise proposed water height after the tsunami event</td>
</tr>
<tr>
<td>Use Case ID</td>
<td>CS1-UC01</td>
</tr>
<tr>
<td>Revision</td>
<td>CS1-UC01-01</td>
</tr>
<tr>
<td>Status</td>
<td>Active</td>
</tr>
<tr>
<td>Goal</td>
<td>To get a map of the affected area with the proposed water height after the tsunami event</td>
</tr>
<tr>
<td>Summary</td>
<td>The user opens the browser which shows map-window with the water height after the tsunami event in the affected area</td>
</tr>
<tr>
<td>Category</td>
<td>primary</td>
</tr>
<tr>
<td>Actor</td>
<td>Employee in a local tsunami warning centre</td>
</tr>
<tr>
<td>Primary Actor</td>
<td>Employee in a local tsunami warning centre</td>
</tr>
<tr>
<td>Stakeholder</td>
<td></td>
</tr>
<tr>
<td>Data input</td>
<td>satellite scene with near infra-red and visible spectrum (e.g. Landsat); bounding box with spatial extent (e.g. WGS84); temporal extent (tmmmmjjjj, hh:mm), calculated forecast of the water height</td>
</tr>
<tr>
<td>Data access control</td>
<td>no special access control</td>
</tr>
<tr>
<td>Data format</td>
<td>digital raster dataset</td>
</tr>
<tr>
<td></td>
<td>image in the browser</td>
</tr>
<tr>
<td>Preconditions</td>
<td>The user has opened the portal successfully</td>
</tr>
</tbody>
</table>
User Story template

I <in the role of XX> needs functionality <zzz> to achieve the goal of <YYY>
Backlog metamodel