TDT4252 / DT8802
Enterprise Modelling and Enterprise Architecture
ArchiMate for Enterprise modelling

John Krogstie
Professor, IDI, NTNU
Overview of presentation

- Recap on enterprise modeling from last week
- Main aspects and roots of ArchiMate
- Business (enterprise) modelling in ArchiMate
- Defined modeling viewpoints on the business layer

- Material: Gerben Wierda. ArchiMate 2.0 Understanding the Basics
Enterprise Models: Definition

- An enterprise model is a consistent set of special-purpose and complementary models describing various facets of an enterprise to satisfy some purpose of some business users. (Vernadat)

- In this case, the purpose of business users mostly deals with describing, designing, analysing, deciding or controlling operations and components of this enterprise.

- The contents of the enterprise model is whatever the business user considers important to describe.
Enterprise Modelling: Purpose

Remember we said that modelling is not always for IT systems design……

- To represent and understand how the enterprise works.
- To capitalise acquired knowledge and know-how for later use.
- To rationalise and secure information.
- To (re)design and specify a part of an enterprise (functional, information, organisational or structural aspects).
- To analyse some aspects of the enterprise (economic analysis, organisational, qualitative, etc.)
- To simulate the behaviour of some parts of the enterprise.
- To make better decisions about enterprise operations and organisation.
- To control, coordinate or monitor some parts of the enterprise.

It’s all about the ENTERPRISE, not just an IT system!
The ArchiMate Research Project

- Develop a modelling language for representing Enterprise Architectures
- 2½ years, July 2002 - December 2004
- approx. 35 man-years, 4 million euro
- Consortium of companies and research institutes
- Directed by Novay (then Telematica Instituut)

Universiteit Leiden  
Radboud Universiteit Nijmegen  
Belastingdienst  
CWI
ArchiMate focus

Visualisation

Analysis

Integration
Main Traits of ArchiMate

- **A Lean language:**
  - just enough concepts, not bloated to include everything possible
  - 80/20 rule

- **Well-founded concepts & models give precision**
  - clear communication about architectures
  - get away from the ‘fuzzy pictures’ image

- **Links to existing approaches**
  - UML, BPMN, TOGAF

- **International vendor-independent standard**
  - The Open Group

- **Tool support**
  - several tools available
Layers, aspects, and viewpoints in ArchiMate

- Information
- Processes
- Business Functions
- Organisation
- Data
- Applications
- Technical Infrastructure

Passive structure: "object"

Behaviour: "verb"

Active structure: "subject"
Short on layers in ArchiMate relative to Enterprise Architecture frameworks and methods

- Zachman Framework
- TOGAF
- NB focus in this lecture is on the business layer (cf. focus in enterprise modeling)
- More on the use of a full Enterprise Architecture and Archimate in connection to this later in the course
## Zachman’s EA Framework

<table>
<thead>
<tr>
<th>Aspects/perspectives</th>
<th>DATA</th>
<th>Function</th>
<th>Time</th>
<th>Motivation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Scope (Contextual)</strong></td>
<td>'Things Important to the Business'</td>
<td>List of Processes the Business Performs</td>
<td>List of Locations in which the Business Operates</td>
<td>List of Organizations Important to the Business</td>
</tr>
<tr>
<td>planner</td>
<td>Ent = Class of Business Thing</td>
<td>Function = Class of Business Process</td>
<td>Node = Major Business Location</td>
<td>People = Major Organizations</td>
</tr>
<tr>
<td>Owner</td>
<td>E.g. Semantic Model</td>
<td>E.g. Business Process Model</td>
<td>E.g. Work Flow Model</td>
<td>E.g. Master Schedule</td>
</tr>
<tr>
<td>Owner</td>
<td>E.g. Logical Data Model</td>
<td>E.g. Application Architecture</td>
<td>E.g. Human Interface Architecture</td>
<td>E.g. Business Plan</td>
</tr>
<tr>
<td>Egress</td>
<td>E.g. Physical Data Model</td>
<td>E.g. Presentation Architecture</td>
<td>E.g. Control Structure</td>
<td>E.g. Rule Design</td>
</tr>
<tr>
<td>ingress</td>
<td>E.g. Data Definition</td>
<td>E.g. Program</td>
<td>E.g. Network Architecture</td>
<td>E.g. Timing Definition</td>
</tr>
<tr>
<td>Sub-Contractor</td>
<td>E.g. Function</td>
<td>E.g. ORGANIZATION</td>
<td>End = Sub-condition Means = Step</td>
<td></td>
</tr>
</tbody>
</table>

---

**John A. Zachman, Zachman International (810) 231-0531**
ArchiMate and Zachman

<table>
<thead>
<tr>
<th>Scope</th>
<th>Enterprise Model</th>
<th>System Model</th>
<th>Technology Model</th>
<th>Detailed representation</th>
<th>Functioning Enterprise</th>
</tr>
</thead>
<tbody>
<tr>
<td>Planner’s view</td>
<td>Owner’s view</td>
<td>Designer’s view</td>
<td>Builder’s view</td>
<td>Subcontractor’s view</td>
<td>User's view</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>What</th>
<th>How</th>
<th>Where</th>
<th>Who</th>
<th>When</th>
<th>Why</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Contextual
Conceptual
Logical
Physical
As Built
Functioning
TOGAF, ArchiMate and Extensions
From lecture on perspectives to conceptual modelling

- **Structural** - Passive aspect. More detailed using e.g. UML class diagrams
- **Functional** - Behaviour aspect
- **Behavioral** - Only limited control flow, more detailed using e.g. BPMN
- **Rule-oriented** - Motivational extension
- **Object-oriented** - Not directly, more details using e.g. UML
- **Social communication** – Speech acts mentioned under passive structure (meaning)
- **Actor/role-oriented** – Active aspect
- **Topological** - Location (as part of the active aspect)
Abstraction Levels

Generic concepts

Enterprise architecture concepts

Company-specific concepts, standards, e.g. UML, BPMN
ArchiMate and UML/BPMN/…

- ArchiMate connects architectural domains
  - Broader scope, but less detail than e.g. UML (software), BPMN (processes)
  - No replacement for these, but an ‘umbrella’ on top
- Several ArchiMate concepts/notation derived from BPMN (esp. business processes) and UML (esp. for application and infrastructure)
  - Easy to link to e.g. UML descriptions of detailed design or BPMN process models
Service Orientation in ArchiMate

Service

- Unit of externally available functionality
- Offered via clear interfaces to the environment
- Hide internal operations

Service Oriented Architecture (SOA) as an example on the Technology layer

Definition

- “A set of components which can be invoked, and whose interface descriptions can be published, discovered and invoked over a network.” (W3C)
Web Services as an example technology to implement SOA

Web services architecture

- Web services can be used to implement service-oriented solutions
- They adhere to the set of roles and operations specified by the service oriented model.
- They have also managed to establish a standardized protocol stack.
Services as binding concept in ArchiMate

Customer

External business service

Internal business service

External application service

Internal application service

External infra. service

Internal infra. service
Enterprise Modelling with ArchiMate

- The ArchiMate modelling language on the Business layer
- Viewpoints on the Business layer
Layers, aspects and viewpoints

- Passive structure: "object"
- Behaviour: "verb"
- Active structure: "subject"
ArchiMate Concepts on the business level using an insurance company example
ArchiMate Notation

- Most concepts have two notations:
  - Icon
  - Box with icon
- Sharp corners = active/passive structure
- Rounded corners = behavioural aspects
- Notation resembles UML and BPMN
- Relations (arrows etc.) are also mostly taken from existing languages
- Colors used e.g. according to aspect or layers (not necessarily used like this in all tools)
Generic meta-model in ArchiMate

- **Service**
  - accessed by
  - realized by
  - triggered by / flow from
  - triggers / flow to
  - assigned from

- **Interface**
  - assigned to
  - composed of
  - used by
  - uses

- **Passive Structure Element**
  - accessed by
  - realizes
  - used by

- **Behavior Element**
  - accessed by
  - assigned from
  - used by
  - uses

- **Active Structure Element**
  - assigned to
  - composed of
  - used by
  - uses
# Relationships

<table>
<thead>
<tr>
<th>Structural Relationships</th>
<th>Notation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Association</td>
<td>Association models a relationship between objects that is not covered by another, more specific relationship.</td>
</tr>
<tr>
<td>Access</td>
<td></td>
</tr>
<tr>
<td>The access relationship models the access of behavioral concepts to business or data objects.</td>
<td></td>
</tr>
<tr>
<td>Used by</td>
<td></td>
</tr>
<tr>
<td>The used by relationship models the use of services by processes, functions, or interactions and the access to interfaces by roles, components, or collaborations.</td>
<td></td>
</tr>
<tr>
<td>Realization</td>
<td></td>
</tr>
<tr>
<td>The realization relationship links a logical entity with a more concrete entity that realizes it.</td>
<td></td>
</tr>
<tr>
<td>Assignment</td>
<td></td>
</tr>
<tr>
<td>The assignment relationship links units of behavior with active elements (e.g., roles, components) that perform them, or roles with actors that fulfill them.</td>
<td></td>
</tr>
<tr>
<td>Aggregation</td>
<td></td>
</tr>
<tr>
<td>The aggregation relationship indicates that an object groups a number of other objects.</td>
<td></td>
</tr>
<tr>
<td>Composition</td>
<td></td>
</tr>
<tr>
<td>The composition relationship indicates that an object is composed of one or more other objects.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dynamic Relationships</th>
<th>Notation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow</td>
<td></td>
</tr>
<tr>
<td>The flow relationship describes the exchange or transfer of, for example, information or value between processes, function, interactions, and events.</td>
<td></td>
</tr>
<tr>
<td>Triggering</td>
<td></td>
</tr>
<tr>
<td>The triggering relationship describes the temporal or causal relationships between processes, functions, interactions, and events.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Other Relationships</th>
<th>Notation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grouping</td>
<td></td>
</tr>
<tr>
<td>The grouping relationship indicates that objects, of the same type or different types, belong together based on some common characteristic.</td>
<td></td>
</tr>
<tr>
<td>Junction</td>
<td></td>
</tr>
<tr>
<td>A junction is used to connect relationships of the same type.</td>
<td></td>
</tr>
<tr>
<td>Specialization</td>
<td></td>
</tr>
<tr>
<td>The specialization relationship indicates that an object is a specialization of another object.</td>
<td></td>
</tr>
</tbody>
</table>
Business layer meta-model
Active concepts on the business level

Business actor

Business role

Business collaboration

Business interface

Location

Main office

Legal department

Finance department

Local office

Claims handling department

Sales support

Sales department

Medical insurance sales

Medical insurance seller

Medical insurance department

Luggage insurance sales

Luggage insurance seller

Luggage insurance department
Behavioural concepts on the business layer
Passive concepts on the business layer

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

- Request insurance → Receive request → Process request → Collect premium
- Request form
- Invoice
- Bill
- Telebanking account
  - Service conditions
  - Service level agreement
Example Business Layer Model

Value

Business actor

Business role

Be insured (security)

Client

Insurant

ArchiSurance

Insurer

Damage insurance

Business service

Insurance policy

Claims registration service

Customer information service

Claims payment service

Damage claiming process

Claim submitted

Registration

Acceptance

Valuation

Payment

Invoice

Business event

Business process

Business function

Contract

Product
Motivation aspects at the business layer
Drivers, assessment and goals

Profit

Sales

Application costs too high

Reduce maintenance costs

Costs

Employee costs too high

Reduce direct application costs

Reduce workload employees

Reduce interaction with customers

Reduce manual work
Goals, requirements and services

Lack of insight in portfolio

Improve portfolio management

Employee costs too high

Reduce workload employees

Reduce interaction with customer

Reduce manual work

Facilitate self-service

Make customer interaction more effective

Assign personal assistant

Provide on-line portfolio service

Provide on-line information service

Personal portfolio service

On-line portfolio service

On-line information service

Assistant

Portfolio management application

Product information application
Viewpoints in ArchiMate

- Limited modelling-palettes based on different concerns of different stakeholders

- Viewpoint classification
  - Designing a new enterprise -> architect, developer
  - Deciding – supporting decision making -> product manager, CIO, CEO
  - Informing -> customer, employee, other stakeholder

- Abstraction level
  - Details – one layer/one aspect
  - Coherence – multiple layers or aspects
  - Overview – both multiple layers and multiple aspects
Predefined viewpoints on the business layer

- Organization viewpoint
- Business function viewpoint
- Business process viewpoint
- Product viewpoint
- Information viewpoint
Organization viewpoint

- Business interface
- Business role
- Business actor
- Location

ArchiSurence
- Front Office
  - Customer Relations
  - Intermediary Relations
- Back Office
  - Home & Away
  - Car
  - Legal Aid
  - Finance
  - Product Development
  - HRM
  - Document Processing SSC
Business function viewpoint – designing, coherence
Business process viewpoint – designing detail
Business process example
Product viewpoint – designing/deciding, coherence
Example of product viewpoint
Information viewpoint - Designing details
Example on information structure
Extension mechanisms

- Additional properties (cf. Tagged values in UML)
- Sub-types (cf. Stereotypes in UML)
Overview of Core Concepts across layers

Business
- Business object
- Business process
- Business service
- Business role
- Business actor

Application
- Data object
- Application function
- Application service
- Application component
- Application interface
- Infrastructure service
- Infrastructure interface

Technology
- Artifact
- System software
- Device
- Network

Passive structure

Behaviour

Active structure
Tool Support for ArchiMate

- **Commercial:**
  - BiZZdesign: Architect
  - Software AG: ARIS ArchiMate Modeler
  - Sparx: Enterprise Architect
  - IBM: System Architect (via Corsa plugin)
  - Casewise: Corporate Modeler
  - Avolution: Abacus
  - Agilense: EA Web Modeler
  - Promis: EVA Netmodeler
  - Visual Paradigm: Agilian
  - MEGA: MEGA for ArchiMate
  - Orbus: iServer Enterprise Architect
  - Troux: Metis (oude versie)

- **Freeware:**
  - Archi (Bolton Univ.) archi.cetis.ac.uk
  - ArchiLe sourceforge.net/projects/archile/
  - Microsoft Visio www.archimate.org

- **Online:**
  - Archivity www.archivity.org
  - ModelWorld www.modelworld.nl
Active User Community

- ArchiMate Forum of The Open Group
  - http://www.archimate.org
- NAF working group
  - http://www.naf.nl/nl/werkgroepen/archimate.html
- ArchiMate LinkedIn group
  - http://www.linkedin.com/groups/ArchiMate-50758
  - > 2000 members
Final questions