Mapping TOGAF® to ArchiMate®
AKA Terminology Torture

Graham Berrisford
Avancier Ltd
Harmonising two conceptual frameworks for EA

Mapping TOGAF® to ArchiMate®

AKA Terminology Torture

Including some slides from Avancier’s training to BCS Enterprise and Solution Architecture Certificates

Copyright © Avancier Ltd, http://avancier.website
What do we want for EA?

What is needed for the architectural design and planning of large-scale changes to business systems?

- Processes
- Products
- People
Processes and products

► **TOGAF** best known for its process
  ■ the Architecture Development Method (ADM)

► Also offers a menu of lightly-defined products
  ■ Deliverables
  ■ Artifacts
    ● Catalogs
    ● Matrices
    ● **Diagrams**

► **ArchiMate** more limited
  ■ a modelling language for drawing **diagrams**

► A perfect marriage?
### TOGAF 9.2 artifact types – **diagram types in the handout**

<table>
<thead>
<tr>
<th>PRELIMINARY</th>
<th>VISION</th>
<th>REQUIREMENTS</th>
<th>PLANNING</th>
</tr>
</thead>
<tbody>
<tr>
<td>Principles Catalog</td>
<td>Stakeholder Catalog</td>
<td>Requirements Catalog</td>
<td>Project Context Diagram</td>
</tr>
<tr>
<td>Driver/Goal/Objective Catalog</td>
<td>Solution Concept Diagram</td>
<td>Requirements Traceability Matrix</td>
<td>Benefits Diagram</td>
</tr>
<tr>
<td>Value Chain Diagram</td>
<td>Business Concept Diagram</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Business Model Diagram</td>
<td><strong>BUSINESS</strong></td>
<td><strong>PROCESS VIEW</strong></td>
<td></td>
</tr>
</tbody>
</table>

**Capability view**

- Goal/Objective/Service Diagram
- Functional Decomposition Diagram
- Business Service/Function Catalog
- Business Interaction Matrix
- Node Connectivity Diagram
- Business Capabilities Catalog
- Strategy/Capability Matrix
- Business Capability Map

**People view**

- Organization Decomposition Diagram
  - Function/Organization Matrix
  - Role Catalog
  - Organization/Role Catalog
  - Actor/Role Matrix
  - Location Catalog
- Organization Map
  - Capability/Organization Matrix
- Value Stream (Stages) Catalog
  - Value Stream/Capability Matrix
  - Value Stream Map

**Process view**

- Process Catalog
  - Business Use Case Diagram
  - Process Flow Diagram
  - Event Diagram
  - Product Lifecycle Diagram

**Business data view**

- Conceptual Data Diagram
- Data Entity/Business Function Matrix
- Business Service/Information Diagram

**DATA**

- Data Entity/Data Component Catalog
- Application/Data Matrix
- Logical Data Diagram
- Data Dissemination Diagram
- Data Security Diagram
- Data Migration Diagram
- Data Lifecycle Diagram

**APPLICATIONS**

- Application Portfolio Catalog
- Interface Catalog
- Application/Func/Org/Role Matrices
- Information Exchange Matrix
- Application Communication Diagram
- Process Application Realization Diagram
- Application and User Locations Diagram
- Application Use Case Diagram
- Application Migration Diagram

**TECHNOLOGY**

- Technology Portfolio Catalog
- Technology Standards Catalog
- Application/Technology Matrix
- Environment Locations Diagram
- Platform Decomposition Diagram

**IMPLEMENTATION**

- Software Engineering Diagram
- Software Distribution Diagram
- Enterprise Manageability Diagram
- Processing Diagram
- Networked Computing Hardware Diagram
- Communications Engineering Diagram

---

Copyright Avancier Ltd 2018
ArchiMate diagram symbols
This presentation

- Not about diagrams!
- See the handout for examples
- Rather, about the *concepts in the diagrams*

- Because communication requires that
  - Speakers and hearers share an understanding of the concepts spoken words represent.
  - Drawers and readers share an understanding of the concepts the diagrams symbols represent.
The perfect marriage?

- The bride and groom come from different families, with different cultures.

- **ArchiMate**
  - tries to present a coherent and consistent conceptual framework
  - using a tightly controlled language.

- **TOGAF**
  - tends to embrace all words and concepts its authors have found useful in practice.
  - leading to incoherence
  - e.g. new authors undermined
    - “Building Block” and “Service” in v9
    - “Function” in v9.2
Nevertheless

- If you know the history of TOGAF, you can find a coherent conceptual framework in it.

- This presentation compares and contrasts
  - the conceptual framework of TOGAF
  - the conceptual framework of ArchiMate.
The initial direction to EA

1. The initial direction to EA
2. What is a business system?
3. Service-orientation in the TOGAF standard
4. Abstraction in TOGAF
5. The generic meta model that underpins ArchiMate
6. Mapping terms in the two standards
7. What is the function/process distinction?
8. Data architecture in TOGAF
9. Mapping the concepts to TOGAF artifacts
10. An alternative Business Architecture approach
Motivations and context for EA

“Business planning at the strategy level provides the initial direction to Enterprise Architecture”. (Ch. 5)
Business direction words used in TOGAF 9.2

- Business Driver
- Mission Statement
- Business Vision
- Business Principle
- Goal
- Strategy
- Objective

Figure 5-3 Interoperability and Relationships between Management Frameworks
In the OMG’s Business Motivation Model

- TOGAF’s Principles are “Directives”
- TOGAF’s Goals and Objectives are “Desired results”
- “Course of action” is a plan, high or low level, to achieve those desired results.

Contents of “Business Direction” in TOGAF 9.2

- Business Driver
- Mission Statement
- Business Vision

- Directive
  - Business Principle

- Desired Result
  - Goal
  - Objective

- Course of Action
  - Strategy

A business strategy document may include principles, drivers and goals, along with very high level plans or road maps.
Adding more words

“Means” and “End” words in the Business Motivation Model from the OMG

- Mission
- Vision
- Directive
- Desired Result
- Course of action
- Strategy
- Tactic
- Goal
- Objective
- Business Policy
- Business Rule

Contents of “Business Direction” in TOGAF 9.2

- Business Driver
- Mission Statement
- Business Vision
- Directive
- Desired Result
- Course of Action
- Business Principle
- Goal
- Strategy
- Project
- Business Policy
- Objective
- Work Package

Copyright Avancier Ltd 2018
ArchiMate has symbols for some of these words (and more)
TOGAF’s primary Business Direction artifact

Driver/Goal/Objective catalog

“a cross-organizational reference... a definitive breakdown of drivers, goals, and objectives... to identify synergies across the organization”
What is a business system?

1. The initial direction to EA
2. **What is a business system?**
3. Service-orientation in the TOGAF standard
4. Abstraction in TOGAF
5. The generic meta model that underpins ArchiMate
6. Mapping terms in the two standards
7. What is the function/process distinction?
8. Data architecture in TOGAF
9. Mapping the concepts to TOGAF artifacts
10. An alternative Business Architecture approach
From business planning to business system planning

“EA structures the business planning into an integrated framework that regards the enterprise as a system or system of systems.” (Ch. 5)

EA is about business system planning.
What is a business system?

► One or more business operations that we can model as

► A discrete event-driven system in which
  ■ events trigger
  ■ active structures (actors) to perform
  ■ behaviors (repeatable activities)
Active structures

- Actor
- Organisation unit
- Component
- Module
- Node

Subsystems or components that can be encapsulated by the services they perform, in response to events.

Barperson services
- Serve customer
- Take Order
- Serve Product
- Take Payment
- Open doors
- Close doors
- Clean up
Service

- A behavior exposed at the interface of a system or component.

- It may
  - update the internal state of the system
    - “add value” to an artifact or activity.
  - consume and produce I/O flows
    - flows of data and sometimes materials
A service is definable by a contract

- A service contract comprises
  - Name
  - Inputs
  - Outputs
  - Rules
  - Quality measures
Services encapsulate processes

- one or more processes are needed to complete a service.

Process flow
- A sequence of sub processes (stages, steps or activities)
- Terminates in the production of a flow or other result of value.

Initiate sale ➔ Discuss needs ➔ Configure products ➔ Verify delivery ➔ Price order ➔ Confirm order ➔ Place Order ➔ Capture signature
Behaviors cannot perform behaviors!

- Two behaviors can collaborate
  - one can invoke the other.

- A behavior cannot *perform* another behavior; you need an
  - actor to perform an activity (aka)
  - active structure to perform a behavior
A passive structure

- A structure that does not act, but is acted on.
- It can be a material or data structure.
Data entities and components in TOGAF

**Data entity**
- A unit of data recognized by a domain expert
- identified with a thing or concept of importance in their domain,
- part of a data component.

**Data component**
- A data structure composed of data entities
Business objects in ArchiMate

► Business Object
- a material thing?
- a data representation of it?
- a data sent/received in a data flow
- a data entity created/used in a data store?
- any of the above?

ArchiMate standard example suggests
- a logical data flow, containing a data structure, which may be conveyed in different physical forms.
Service-orientation in the TOGAF standard

1. The initial direction to EA
2. What is a business system?
3. **Service-orientation in the TOGAF standard**
4. Abstraction in TOGAF
5. The generic meta model that underpins ArchiMate
6. Mapping terms in the two standards
7. What is the function/process distinction?
8. Data architecture in TOGAF
9. Mapping the concepts to TOGAF artifacts
10. An alternative Business Architecture approach
The Open Group’s general principle

- Standards should be based on “executable specification”.

- Specify a system by defining the behaviors it is required to perform, and their results.

- TOGAF applies this principle to the specification of business systems as well as IT systems.

AutoXpress Services
- Fit tyres
- Check-up and oil change
- Full annual service
- Check brakes
- Repair brakes
- Check exhaust
- Replace exhaust
- Inspect battery
- Replace battery
- Align wheels
- Replace windscreen wipers
- Fit bulbs
- Replace shock absorbers
The unit of behavior - a service

► a required behavior or unit of work

► defined as a service requester sees it.
  ■ hides how a system works.

► may be short or long
  ■ depending on what the requester wants and the resources available to the service provider.

► realised by one or more components performing one or more processes.

AutoXpress Services

➢ Fit tyres
➢ Check-up and oil change
➢ Full annual service
➢ Check brakes
➢ Repair brakes
➢ Check exhaust
➢ Replace exhaust
➢ Inspect battery
➢ Replace battery
➢ Align wheels
➢ Replace windscreen wipers
➢ Fit bulbs
➢ Replace shock absorbers
The unit of structure - a “building block” (or component)

- a subsystem or actor of any kind, including human and computer actors.
- may be coarse-grained or fine-grained
- can realise one or more services
  - on its own or in collaboration with others
- can play the
  - client role of service requester and/or
  - server role of service provider.
Service or Building Block? Which is bigger?

- Services may be long or short
  - One long service (e.g. package delivery) may require the participation of very many BBs.
The trouble with “building block” and “service”

- Changes in successive versions of the standard have left the terms ambiguous.

- In this presentation, building block and component are synonyms.

- Service is an external view of processes.
Abstraction in TOGAF

1. The initial direction to EA
2. What is a business system?
3. Service-orientation in the TOGAF standard
4. Abstraction in TOGAF
5. The generic meta model that underpins ArchiMate
6. Mapping terms in the two standards
7. What is the function/process distinction?
8. Data architecture in TOGAF
9. Mapping the concepts to TOGAF artifacts
10. An alternative Business Architecture approach
## Abstraction in ArchiMate and TOGAF

<table>
<thead>
<tr>
<th>Delegator</th>
<th>Composition</th>
<th>Generalisation</th>
<th>Idealisation</th>
</tr>
</thead>
<tbody>
<tr>
<td>a client element</td>
<td>a composite or aggregate element</td>
<td>a generalised element</td>
<td>an idealised or logical element</td>
</tr>
<tr>
<td>server element(s)</td>
<td>smaller or shorter elements</td>
<td>particular subtypes</td>
<td>“physical” features</td>
</tr>
</tbody>
</table>

### Abstraction Types
- **Delegation**: Indicates a client element delegating a request to a server element(s).
- **Composition**: Indicates a composite or aggregate element composed of smaller or shorter elements.
- **Generalisation**: Indicates a generalised element representing particular subtypes.
- **Idealisation**: Indicates an idealised or logical element representing “physical” features.

### Abstraction in Business Applications and Technologies
- **Business Applications**: Understand business needs and requirements.
- **Technologies**: Explore technology solutions and capabilities.
- **Enterprise Segments**: Consider enterprise segments and their solutions.
- **Common System**: Focus on common system components.
- **Industry Organisation**: Analyse industry-specific organisational structures.
- **Services**: Examine services and their integration.
- **Logical ABBs**: See logical architectural building blocks.
- **Physical SBBs**: Identify physical architectural building blocks.
- **Deployed Solutions**: Implement and deploy solutions.
Abstraction by generalisation of system description

Generic

- Generic
- Widespread
- Parochial
- Specific

Specific

- Universal
- Common
- Domain specific
- Unique
- Foundation
- Common system
- Industry
- Organisation

TOGAF
<table>
<thead>
<tr>
<th>Idealised description</th>
<th>Executable description</th>
<th>Realisation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conceptual</td>
<td>Computation Independent Model</td>
<td>Executable Artifact</td>
</tr>
<tr>
<td>Logical</td>
<td>Platform Independent Model</td>
<td>Executing Artifact</td>
</tr>
<tr>
<td>Physical</td>
<td>Platform Specific Model</td>
<td>Configuration</td>
</tr>
<tr>
<td></td>
<td>Executable Artifact</td>
<td>Instantiation</td>
</tr>
<tr>
<td></td>
<td>Realisation</td>
<td>Deployed Solutions</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Common</th>
<th>OMG MDA</th>
<th>Zachman</th>
<th>TOGAF</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Abstraction by idealisation of system description

- Common
- OMG MDA
- Zachman
- TOGAF

Copyright Avancier Ltd 2018
# TOGAF’s Enterprise Continuum

<table>
<thead>
<tr>
<th>Generalisation</th>
<th>Foundation</th>
<th>Common system</th>
<th>Industry</th>
<th>Organisation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Idealisation</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Requirements and context</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Service Contracts</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Architecture continuum</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Architecture BBs</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Solution continuum</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Solutions BBs</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Deployed solutions</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Deployed Solutions</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Level of idealisation in TOGAF’s Enterprise Continuum

<table>
<thead>
<tr>
<th>Idealisation</th>
<th>Generalisation</th>
<th>Architecture Requirements Specification includes Business and Application/IS <strong>Services contracts.</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements and context</td>
<td>Service Contracts</td>
<td>Logical Components are defined by the services they provide to each other and to external entities, also by the abilities and data resources they need.</td>
</tr>
<tr>
<td>Architecture continuum</td>
<td>Architecture BBs</td>
<td>Physical Components are procurable; they can be hired, bought or built to realise Logical Components, and so deliver the required Services.</td>
</tr>
<tr>
<td>Solution continuum</td>
<td>Solutions BBs</td>
<td>Real world components are employed or deployed to do work at run-time (might be called Operational Components).</td>
</tr>
<tr>
<td>Deployed solutions</td>
<td>Deployed Solutions</td>
<td></td>
</tr>
</tbody>
</table>
The content framework is based on a generic relation

### Idealised description

<table>
<thead>
<tr>
<th>Services</th>
<th>Logical Components</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;are clustered and assigned to&gt;</td>
<td>Logical: “An implementation-independent definition”, portable and supplier-neutral.</td>
</tr>
<tr>
<td>&lt;are realised by&gt;</td>
<td>Physical: “A description of a real-world entity”, still “considerably abstracted from implementation”.</td>
</tr>
<tr>
<td>Physical Components</td>
<td></td>
</tr>
<tr>
<td>&lt;are instantiated as&gt;</td>
<td></td>
</tr>
<tr>
<td>Deployed solutions</td>
<td></td>
</tr>
</tbody>
</table>

### Realisation
## TOGAF generic relation

<table>
<thead>
<tr>
<th>Business Service/ Function catalog</th>
<th>Services</th>
<th>&lt;are clustered and assigned to&gt;</th>
<th>Logical Components</th>
<th>&lt;are realised by&gt;</th>
<th>Physical Components</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business Service/ Function catalog</td>
<td>Business Services</td>
<td>Logical Components</td>
<td>Functions</td>
<td>Physical Components</td>
<td></td>
</tr>
<tr>
<td>Role catalog + Actor/Role matrix</td>
<td>Activities</td>
<td>Logical Components</td>
<td>Roles</td>
<td>Physical Components</td>
<td></td>
</tr>
<tr>
<td>Application portfolio catalog</td>
<td>IS Services</td>
<td>Logical Components</td>
<td>Logical Application Components</td>
<td>Physical Application Components</td>
<td></td>
</tr>
<tr>
<td>Technology portfolio catalog</td>
<td>Technology Services</td>
<td>Logical Components</td>
<td>Physical Technology Components</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Copyright Avancier Ltd 2018
The service to component relationship

- TOGAF encourages architects
- to assign the responsibility for one service to one component and
- to minimise duplication of service provision by different components.

<table>
<thead>
<tr>
<th>N Business Services</th>
<th>&lt;are clustered and assigned to&gt;</th>
<th>1 Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>N IS Services</td>
<td>&lt;are clustered and assigned to&gt;</td>
<td>1 Logical Application Component</td>
</tr>
<tr>
<td>N Technology Services</td>
<td>&lt;are clustered and assigned to&gt;</td>
<td>1 Logical Technology Components</td>
</tr>
</tbody>
</table>

- However, a component can delegate work to other components. So,
- one component may perform many services, and
- one service may be performed by many components.
The logical component to physical component relationship

In the application and technology domains, the ideal is a 1-to-1 relation.

<table>
<thead>
<tr>
<th>1 Logical Application Component</th>
<th>IDEALLY realised by</th>
<th>1 Logical Application Component</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Logical Technology Components</td>
<td>IDEALLY realised by</td>
<td>Logical Technology Component1</td>
</tr>
</tbody>
</table>

In practice, the relationships may be more complex, or logical components may be reverse-engineered to keep the relationship simple.
The Logical/Physical distinction in the business domain

- Functions are logical organization units
- Organization units realize Functions

- Roles are logical Actors
- Actors realize Roles
Strictly, Architecture BBs don’t perform Processes or deliver Services. They specify Solution BBs that can do those things, and realize the Architecture BBs.
The general approach in TOGAF

Baseline analysis abstracts *performed services* from building blocks

Target design starts from the *required services*

Identify Goals and Objectives  Identify Required Services  Map Services to Logical ABBs  Map Logical ABBs to Physical SBBa

Identify Performed Services  Study Physical SBBs

Define Business Services  Define IS Services  Define Technology Services  Define Data Entities

Map Services to Functions  Map Functions to Organization Units  Map Logical App Components to Physical ones  Map Logical Tech Components to Physical ones

Map Services to Logical App Components  Map Services to Logical Data Components  Map Logical Data Components to Physical ones
The generic meta model that underpins ArchiMate

1. The initial direction to EA
2. What is a business system?
3. Service-orientation in the TOGAF standard
4. Abstraction in TOGAF
5. The generic meta model that underpins ArchiMate
6. Mapping terms in the two standards
7. What is the function/process distinction?
8. Data architecture in TOGAF
9. Mapping the concepts to TOGAF artifacts
10. An alternative Business Architecture approach
Archimate generic meta model

<table>
<thead>
<tr>
<th>External requirements of external entities</th>
<th>Behaviour</th>
<th>Structure</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>what the system does</td>
<td>what the system is made of</td>
</tr>
</tbody>
</table>

- **External Behavior Element (Service)**
  - accesses
  - realizes
  - serves
  - assigned to

- **External Active Structure Element (Interface)**
  - composes
  - serves

- **Internal Behavior Element**
  - triggers / flows to
  - assigned to

- **Internal Active Structure Element**
  - serves

- **Event**
  - triggers / flows to
  - assigned to
Archimate generic meta model - simplified

<table>
<thead>
<tr>
<th>External</th>
<th>Behaviour what the system does</th>
<th>Structure what the system is made of</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service: an explicitly defined exposed behavior</td>
<td>Interface: a point of access where one or more services are provided to the environment</td>
<td></td>
</tr>
</tbody>
</table>

- **External**: requirements of external entities
  - **Data**
  - **Service**
  - **Process or Interaction**
  - **Service Portfolio**
  - **Component**

- **Internal**: the workings of the system
  - **Accesses**
  - **Realizes**

**Definitions**

- **Data**: A unit of activity performed by one or more active structure elements (e.g. process).
- **Service**: An entity that is capable of performing behavior (e.g. component).
Recursive architecture description

► Both building blocks and services are recursively composed

► One building block perform many services

► One service may require many building blocks (be they nested or sequential).
Mapping terms in the two standards

1. The initial direction to EA
2. What is a business system?
3. Service-orientation in the TOGAF standard
4. Abstraction in TOGAF
5. The generic meta model that underpins ArchiMate
6. **Mapping terms in the two standards**
7. What is the function/process distinction?
8. Data architecture in TOGAF
9. Mapping the concepts to TOGAF artifacts
10. An alternative Business Architecture approach
# Mapping ArchiMate to General System Theory

## Passive Structure | Behavior | Active Structure | GST
--- | --- | --- | ---
Data Entity | Service | Service Portfolio | External
Process | Create & Use | Deliver | Internal
Component | Perform | An encapsulated package of capability

---

## Passive Structure | Behavior | Active Structure | ARCHIMATE
--- | --- | --- | ---
Passive Structure | Service | Interface | External
Process OR Function | Deliver | Realized by | Internal
Active Structure | An encapsulated package of capability
## Mapping TOGAF to ArchiMate

### Passive Structure | Behavior | Active Structure | TOGAF
---|---|---|---
Data Entity | Process | Service Portfolio | External

### Passive Structure | Behavior | Active Structure | ARCHIMATE
---|---|---|---
Passive Structure | Process OR Function | Interface | External

### Active Structure
- **Service Portfolio**
- **Architecture Building Block**
- **Solution Building Block**
- **Active Structure**
- **Interface**
- **Passive Structure**

### Logical
- A logical specification of below
- An encapsulated package of capability

### Physical
- An encapsulated package of capability

---

Copyright Avancier Ltd 2018
ArchiMate’s structure/behavior distinction is peculiar

The standard examples and users often apply the Function symbol to a Process (which can confuse, since Functions are more like Roles than Processes).
Label Logical Components using ArchiMate’s “Function” symbol?

OK. Provided this caveat is understood

- ArchiMate standard and users use the Function symbol for a Process instead
- Logical components in TOGAF (defined by service portfolios) might be seen as closer to ArchiMate’s Interface.
Mapping ArchiMate to TOGAF

TOGAF

Business Component
  - Function
  - Role
  - Actor
  - Organization Unit

Process
  - Uses IS Service
  - Realized by IS Service
  - Create and use Data Entity
  - Realized by Data Entity
  - Held in Technology Service
  - Realized by Technology Service

Application Component
  - Logical App Component
  - Physical App Component

Application Component
  - Logical App Component
  - Physical App Component

Data Component
  - Logical Data Component
  - Physical Data Component

Data Component
  - Logical Data Component
  - Physical Data Component

Technology Component
  - Logical Tech Component
  - Physical Tech Component

Technology Component
  - Logical Tech Component
  - Physical Tech Component

ArchiMate

Business Service

Process
  - Perform App Service
  - Realized by App Service
  - Maintain Data Object
  - Realized by Data Object
  - Held in Technology Service
  - Realized by Technology Service

Application Component
  - Application Function
  - App Component

Application Component
  - Application Function
  - App Component

Data Object

Data Object

Node

Node

Copyright Avancier Ltd 2018
## Terminology torture – aargh!

<table>
<thead>
<tr>
<th>Generic</th>
<th>Behavior Units</th>
<th>Logical Active Structures</th>
<th>Physical Active Structures</th>
</tr>
</thead>
<tbody>
<tr>
<td>Human</td>
<td>Activities</td>
<td>Roles</td>
<td>Actors</td>
</tr>
<tr>
<td>TOGAF</td>
<td><strong>Services</strong></td>
<td>Logical Components</td>
<td>Physical Components</td>
</tr>
<tr>
<td>UML</td>
<td>Operations</td>
<td>Interfaces or <strong>Services</strong></td>
<td>Components</td>
</tr>
<tr>
<td>WSDL</td>
<td>Operations</td>
<td>Web <strong>Services</strong></td>
<td>Components</td>
</tr>
<tr>
<td>Fashion</td>
<td>Operations</td>
<td>APIs</td>
<td>Micro<strong>Services</strong></td>
</tr>
</tbody>
</table>
What is the function/process distinction?

1. The initial direction to EA
2. What is a business system?
3. Service-orientation in the TOGAF standard
4. Abstraction in TOGAF
5. The generic meta model that underpins ArchiMate
6. Mapping terms in the two standards
7. What is the function/process distinction?
8. Data architecture in TOGAF
9. Mapping the concepts to TOGAF artifacts
10. An alternative Business Architecture approach
A 3 level Functional Decomposition Structure

Strategic management functions
- Strategy
- Fiscal and accounting
- Risk and Compliance
- Performance

Operational functions
- Products
  - Product definition
  - Product engineering
- Marketing
  - Market development
  - Campaigning
- Sales
  - Distribution channels
  - Sales execution
- Customer care
  - Customer service
  - Customer relations
  - Service channel
  - Customer data
- Assets
  - Investment
    - Investment performance
  - Asset inventory
- Money
  - Banking
  - Accounts
  - Cash flow
  - Money market
- Claims
  - Contracts life cycle
  - Claim settlement
  - Contract admin.
  - Claim admin.

Support functions
- Organization
- HR
- Process
- Office
- Facility
- ITSM

This diagram uses ArchiMate symbols

Copyright Avancier Ltd 2018
A process – sequencing atomic activities in a flow

Sales roles

Sales function

- Initiate sale
- Discuss needs
- Configure products
- Verify delivery
- Price order
- Confirm order
- Place Order
- Capture signature

Supply function

- Confirm availability of stock

Delivery function

- Assign resources

Copyright Avancier Ltd 2018
### Mapping atomic activities to 2\textsuperscript{nd} level nodes of a functional decomposition

<table>
<thead>
<tr>
<th>Strategic management functions</th>
<th>Vision and Strategy</th>
<th>Finance</th>
<th>Product Design</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Operational functions</strong></td>
<td>Sales &amp; Marketing</td>
<td>Supply</td>
<td>Manufacture</td>
<td>Delivery</td>
</tr>
<tr>
<td>Advertise</td>
<td>Select suppliers</td>
<td>Assign resources</td>
<td>Assign resources</td>
<td>Deliver products</td>
</tr>
<tr>
<td>Identify prospect</td>
<td>Purchase goods</td>
<td>Deliver products</td>
<td>Deliver products</td>
<td>Install products</td>
</tr>
<tr>
<td>Initiate sale</td>
<td>Receive goods</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Discuss needs</td>
<td>Inspect goods</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Configure products</td>
<td>Return goods</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Verify delivery</td>
<td>Despatch goods</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Price order</td>
<td>Confirm availability of stock</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Confirm order</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Place Order</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Support functions</strong></td>
<td>Human Resources</td>
<td>Accounts</td>
<td>Facilities</td>
<td>Knowledge and Change</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Legal</td>
<td></td>
</tr>
</tbody>
</table>

Copyright Avancier Ltd 2018
The Function/Process distinction

A function is a package of capability
It can be bounded by a service portfolio

A process flows over time from start to end.
It may be
► encapsulated by one service.
► one of several processes needed to deliver one service.
► contribute to the delivery of more than one service
Data architecture in TOGAF

1. The initial direction to EA
2. What is a business system?
3. Service-orientation in the TOGAF standard
4. Abstraction in TOGAF
5. The generic meta model that underpins ArchiMate
6. Mapping terms in the two standards
7. What is the function/process distinction?
8. **Data architecture in TOGAF**
9. Mapping the concepts to TOGAF artifacts
10. An alternative Business Architecture approach
Data architecture in TOGAF

► Data architecture is about
  ■ data at rest (in store)
  ■ data in motion (in flows).

► What TOGAF classifies under data architecture are artifacts that record the creation and use of data entities in persistent data components.

<table>
<thead>
<tr>
<th>Data entity/data component catalog</th>
<th>Data entities</th>
<th>&lt;are related together in&gt; Logical Data Components</th>
<th>&lt;are realised by&gt; Physical Data Components</th>
</tr>
</thead>
</table>

► What about data flows?
The **Flow** concept implicit in TOGAF artifacts

► **Service**
  - “can be defined in a logical service contract that defines input and output flows and/or state changes.”

► **Flow**
  - A movement of a data and/or material structure between sender and receiver components.

- Q) What about flows that architects choose not specify in service contracts?
Services v data flows in architecture artifacts

► Different concepts, related, but difficult to combine in an artifact
Mapping the concepts to TOGAF artifacts

1. The initial direction to EA
2. What is a business system?
3. Service-orientation in the TOGAF standard
4. Abstraction in TOGAF
5. The generic meta model that underpins ArchiMate
6. Mapping terms in the two standards
7. What is the function/process distinction?
8. Data architecture in TOGAF
9. Mapping the concepts to TOGAF artifacts
10. An alternative Business Architecture approach
Remember terminology torture

- Cap Gemini call services building blocks
- Some call building blocks services
A service-oriented view of business building blocks

- **Driver**: Stimulates the desired result.
- **Goal**: Desired outcome, decomposed into objectives.
- **Objective**: Met by business services.
- **Business Service**: Delivered by components.
- **Business Component or Package of Capability**: Decomposed into functions, roles, organization units, and actors.

Objectives met by services performed by components.
The start of a Business Architecture artifact map

Driver/Goal/Objective catalog

Goal/Objective/Service diagram

Contract Measure catalog

Business Service/Function catalog

Functional Decomposition diagram

Function/Organization matrix

Organization Decomposition diagram

Driver

Goal

Objective

Business Service

Business Component
Or Package of Capability

Function

Role

Organization Unit

Actor

Stimulate

Decomposed into

BMM Desired Result

Delivered by

Met by

Copyright Avancier Ltd 2018
A BA artifact map with Capability and Value Stream artifacts?

- Driver
  - Stimulate
  - BMM
  - Desired
  - Result
- Goal
  - Decomposed into
- Objective
  - Met by
  - Business
  - Service
  - Delivered by
- Capabilities
  - Realized by
    - Business Capability?
    - Business Capability?
    - Role
    - Perform
  - Organization Unit
    - Realized by
  - Actor
    - Realized by
- Process
- Value Stream
- Strategy/Capability Map?
- Business Capability Map?
- Business Capability/ Organization matrix?
- Organization Map?

Copyright Avancier Ltd 2018
The same BA artifact map after some of the draft CRs

Driver/Goal/Objective catalog
Goal/Objective/Service diagram
Business Service/Function catalog
Functional Decomposition diagram
Function/Organization matrix
Organization Decomposition diagram

Business Component Or Package of Capability

Business Interaction matrix
Business Interaction diagram
Business Flow catalog

Role catalog
Actor/Role matrix

Process catalog
Process Flow diagram
Business Event diagram

Copyright Avancier Ltd 2018
A possible solution design sequence

Presuming the baseline organisation structure and functional decomposition are a given
Read the associated paper for further discussion of:

- **In TOGAF**
  - Architecture v solution building blocks
  - The generic relation

- **In ArchiMate**
  - Service v Process (same thing in IAF)
  - Service v Interface (same thing in UML)
  - Actor v Role (same thing in UML)
  - Process v Function (same thing in some sources)
  - Structure v Behavior (peculiar in ArchiMate)
  - Data Object v Business Object

Communication requires that:

Speakers and hearers share an understanding of the concepts spoken words represent.

Drawers and readers share an understanding of the concepts diagrams symbols represent.
An alternative Business Architecture approach

1. The initial direction to EA
2. What is a business system?
3. Service-orientation in the TOGAF standard
4. Abstraction in TOGAF
5. The generic meta model that underpins ArchiMate
6. Mapping terms in the two standards
7. What is the function/process distinction?
8. Data architecture in TOGAF
9. Mapping the concepts to TOGAF artifacts
10. An alternative Business Architecture approach
Capability-oriented artifacts

► **Business Capabilities Catalog**
  ■ A definitive listing of particular abilities that a business may possess or exchange to achieve a specific **purpose**.

► **Business Capability Map**
  ■ A family of diagrams representing a definitive listing of the particular abilities that a business may possess or exchange to achieve a specific purpose.

► **Strategy/Capability Matrix**
  ■ The purpose of this matrix is to show the capabilities required to support specific strategy statements.

► **Organization Map**
  ■ A diagram showing the relationships between the primary entities that make up the enterprise, its partners, and stakeholders.

► **Capability/Organization Matrix**
  ■ The purpose of this matrix is to show the organization elements that implement each capability. The Capability/Organization matrix includes the following metamodel entities:
    ● Business Capability, Value Stream, Organization Unit
About Capabilities and Purposes

► In the BMM, **Desired Result** is the catch all for **Goals** and **Objectives**
► In TOGAF, **Business Drivers > Goals > Objectives > Services**

► How does a Capability’s **Purpose** relate to the above?
► Can you have a Purpose without a Capability? Or are they 1 to 1?
About Capabilities and Building Blocks

► To perform its required behaviors, does every Building Block need a Capability?
  ■ If they are not 1 to 1, why not?
  ■ Where are Capabilities recorded?

► To perform its required behaviors, does every Business Function need a Business Capability?
  ■ If they are not 1 to 1, why not?
  ■ How does a Business Capability Map differ in purpose or use from a Functional Decomposition diagram?
Value stream-oriented artifacts

- **Value Stream Catalog**
  - A definitive listing of end-to-end collections of value-adding activities that create an overall result for a customer, stakeholder, or end user.

- **Value Stream Stages Catalog**
  - A definitive listing of end-to-end collections of the different stages for the value-adding activities that create an overall result for a customer, stakeholder, or end user; it includes the following metamodel entities:
    - Business Capability
    - Value Stream

- **Value Stream/Capability Matrix**
  - The purpose of this matrix is to show the capabilities required to support each stage of a value stream.

- **Value Stream Map**
  - A family of diagrams representing a definitive listing of end-to-end collections of value-adding activities that create an overall result for a customer, stakeholder, or end user.
  - The Value Stream map includes the following metamodel entities:
    - Business Capability
    - Value Stream
About Value Streams and Results

- In the BMM, **Desired Result** is the catch all for **Goals** and **Objectives**
- In TOGAF, **Business Drivers > Goals > Objectives > Services**

- How does a Value Stream’s **Result** relate to the above?.
- Is a Value Stream’s **Result**
  a) the **exit condition** of the process (output flows and system state changes)?
  b) the desired outcome of those being used by some actor to meet some **goal** or **objective**?

- Can a Value Stream Stage be further decomposed?
- Does the Value Stream/Capability Matrix map to all Capabilities, or only to Business Capabilities?
TOGAF does not prescribe which artifacts to produce
Nor the sequence to produce them
But the artifacts do connect together
Alternative business architecture approaches

► BA approach 1

► BA approach 2
Footnotes

► Left overs
Application/Function Matrix drawn as a diagram

Strategic management functions

Vision and Strategy

Operational functions

Sales & Marketing
- Sales and Marketing
- Product Configurator
- Order Entry
- Pricing
- Billing
- Commissions

Supply
- Supply Chain Planning
- Purchasing
- Supplier Scheduling
- Inspection of goods
- Inventory

Manufacture
- Manufacturing Projects
- Manufacturing Process
- Manufacturing Flow
- Bills of Material
- Cost Management
- Quality Control

Delivery
- Scheduling
- Activity Management
- Workflow Management
- Time and Expenses
- Capacity

Customer Service
- Customer Contact
- Call Center support
- Service

Support functions

Human Resources
- Human Resources
- Benefits
- Payroll
- Rostering
- Time and Attendance

Accounts
- Accounts Receivable
- Accounts Payable
- General Ledger
- Fixed Assets
- Cash Management

Facilities

Knowledge and Change
- Training
- Project Management
- Doc Management

Legal

ITSM
- Identity management
- IT Service Management
- Server Management
- Network Management
- EAI Middleware

Finance

Product Design
- Costing
- Engineering

Operations
- Business Intelligence
- Data Warehouse

Copyright Avancier Ltd 2018
A Business Scenario (after TOGAF 8 example)

Roles
(human actors)

Salesman
Customer

Process

Initiate → Discuss → Configure → Verify → Price → Confirm → Place Order → Capture signature

Applications
(computer actors)

Product configurator
Inventory
Scheduling
Pricing
Order Entry

Copyright Avancier Ltd 2018
“Service Portfolio”

- “a collection of services, potentially an interface definition.

- “used in the TOGAF framework to define the requirement for a building block or system.” (Ch. 3)

- “For each building block, build up a service description portfolio as a set of non-conflicting services.”

AutoXpress Services

- Fit tyres
- Check-up and oil change
- Full annual service
- Check brakes
- Repair brakes
- Check exhaust
- Replace exhaust
- Inspect battery
- Replace battery
- Align wheels
- Replace windscreen wipers
- Fit bulbs
- Replace shock absorbers
Subdividing a system into subsystems

- The service portfolio of a system may be divided into service bundles assignable to different subsystems.

- For Portability, Interoperability and Boundaryless Information Flow™.
  - “An architectural framework.. should describe a method for designing an information system in terms of a set of building blocks, and for showing how the building blocks fit together.” TOGAF 7

- For IT architecture.
  - “The TOGAF Technical Reference Model ... contains all possible services.
  - Service bundles are represented .. in the form of "Building Blocks".
  - The IT architect must analyse the services actually needed [to] define the set of optimal solution building block. TOGAF 7
“Systems are built up from collections of building blocks” (Ch. 33)

► ... building blocks have to interoperate with other building blocks.”

• “An architecture is a set of building blocks depicted in an architectural model,
• and a specification of how those building blocks are connected to meet the overall requirements of the business.”
“Building block” in TOGAF chapter 33

► “has a defined boundary”
  ■ Is encapsulated by IO flows

► “recognizable as "a thing" by domain experts”
  ■ Is a structure rather than a transient behavior

► “may interoperate.”
  ■ Building blocks cooperate in a network

► “A good building block
  ■ considers implementation and usage, and
  ■ evolves to exploit technology and standards
    • Is logical, but not divorced from physical reality
  ■ may be assembled from other building blocks, and a subassembly of others
    • may be composed and decomposed in a hierarchical structure
  ■ is re-usable and replaceable, and well specified.”
    • Is a plug and play component
“Building block” in TOGAF chapter 3

- "a (potentially re-usable) component of enterprise capability
- can be combined with other building blocks to deliver architectures and solutions."

- The granularity varies from situation to situation.

- A service can be
  - "coarse-grained (build a house) or"
  - "fine-grained (retrieve an address)." (Ch. 3)
Abstraction by idealisation

“A building block’s boundary and specification should be loosely coupled to its implementation.” (Ch. 33)

“It should be possible to realize a building block in several different ways without impacting [its] boundary or specification.” (Ch. 33)
“The major work… consists of identifying the architecture building blocks required to meet the business goals and objectives.

“The selected set of architecture building blocks is then refined in an iterative process….

to arrive at a set of solution building blocks which can either be bought off-the-shelf or custom developed.” (Ch. 33)

IOW: you hire, buy or build physical components to perform the required behaviors assigned to the logical components.
Abstraction by generalisation

- Resource
  - Material Entity
    - Energy
    - Human
  - Machine
    - Computer
    - Client Device
    - Lap Top
  - Data Structure
  - Other Object
  - Vehicle
  - Server Device

TOGAF
- Foundation
- Common System
- Industry
- Organisation
Note

- **Structures perform behaviors**
  - Roles perform Processes? Hmm...
  - Functions perform Services? Hmm...
- **Logical structures only specify what can perform behaviors**

- **Behaviors do not perform behaviors**
  - Processes do not perform processes.
  - Services do not perform services.
- **But processes can communicate with each other and access data**