TOGAF’s generic conceptual framework with ArchiMate symbols

For more detail about Business Architecture in particular click on this link
http://grahamberrisford.com/00EAframeworks/03TOGAF/TOGAF%20Business%20Architecture%20-%20with%20ArchiMate%20symbols.pdf

In modelling the actors and activities of human and computer activity systems, there are only a few abstract concepts, but for these we have a wide variety of words. If our profession is to adopt a controlled vocabulary, then we have to pin particular words to particular concepts, rather than use words variously and interchangeably.
Requirements <are for> Behaviors.
   e.g. Services, Processes, Value Streams, Scenarios
Behaviors <are assigned for performance to> Logical Building Blocks.
   e.g. Logical Components, Roles, Functions/Capabilities
Logical Building Blocks <are realised by hiring, buying or building> Physical Building Blocks
   e.g. Physical Components, Actors, Organisation units
Common attributes of generic entities

Behaviors

- e.g. Services, Processes, Value Streams, Scenarios
- name
- start conditions: triggering events, inputs, preconditions for success
- end conditions: outputs, results or values delivered, other post conditions
- duration
- volume or throughout
- performers (building blocks)

Inputs and outputs can include materials/goods and information/data flows.

Preconditions and post conditions can refer to data entities accessed or created.

Logical Building Blocks

- e.g. Logical Components, Roles, Functions/Capabilities
- name
- provided interface(s) that group services offered
- required interface(s) that group services needed
- behaviors performed
- physical building blocks that implement the logical building block
- generalised requirements for some or all of the more physical attributes below.

Physical Building Blocks

- e.g. Physical Components, Actors, Organisation units
- name
- deployment place(s) or location(s)
- abilities possessed: power and memory, or skills and knowledge
- physical resources used: materials and/or energy
- logical building blocks realised
Generic concept definitions
Copyright Avancier Limited March 2016 http://avancier.website page 3 of 16

- **Aim:** A statement of need to which compliance can demonstrated. It should be SMART (unlike a principle, which is a more abstract directive or guideline.) Related terms: goal, objective, requirement.

- **Behavior element:** A unit of activity that is triggered by an event and (provided its preconditions are met) terminates with the production of an output or other response. Related terms: service, process, use case, user story.
  - **Service:** A discrete stimulus-response behavior, specified in a service contract that encapsulates process(es) needed to realise the service.
  - **Process:** A step-by-step procedure that realises one or more services, specified as a sequence of activities performed by one or more components. Related terms: scenario, workflow, value stream.

- **Active structure element:** A component of an activity system, responsible for performing behaviors, which may be specified at logical or physical levels.
  - **Logical component:** A structure element specified in terms of services provided and/or processes performed and abilities required. Related terms: interface, function, capability, role.
  - **Physical component:** A structure element that can be hired, bought or built to realise a logical component and perform required behaviors. Related terms: component, organisation unit, actor.
The Open Group has always adopted a **Service-Oriented Approach** to system specification. It presumes active structure elements are or can be encapsulated by (interfaces composed of) discretely invokable "services". So the TOG principle (the modern practice) is to crystallise requirements in Service definitions.

**TOGAF’s generic conceptual framework**

- SMART Requirements
- Logical Design
- Physical Design

Goal, Objective, Requirement

Behavior Element

Active Structure Element

Service

Process

Logical Component

Physical Component

**TOGAF assigns a “Service Portfolio” to a Logical Component (cf. a Function, or Interface, in ArchiMate).**
TOGAF’s general approach to enterprise transformation

TOGAF documents a baseline by defining current Components, then abstracting the Services they deliver.

- Define Services clients require (and sometimes, Processes to deliver Services)
- Cluster Services into “Service portfolios”
- Assign each Service portfolio to a Logical Component,
- Hire, buy or build one or more Physical Components to realise a Logical Component.
Enterprise architecture transformation

Baseline-to-target gap analysis informs the development of:

- Business change road map
- Application change road map
- Technology change road map

Reverse engineer the baseline architecture

Forward engineer the target architecture

Goals, Objectives, Requirements

Business architecture

- Business Service
- Process
- Function, Role
- Org Unit, Actor

Applications architecture

- IS (App) Service
- Logical Application Component
- Physical Application Component

Technology architecture

- Platform Service
- Logical Technology Component
- Physical Technology Component

Copyright Avancier Limited March 2016 http://avancier.website
<table>
<thead>
<tr>
<th><strong>Core terms: defined to fit the conceptual framework</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Copyright Avancier Limited March 2016 <a href="http://avancier.website">http://avancier.website</a> page 8 of 16</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>External entity</strong></th>
<th>[A logical or physical component] outside the business or system of interest, which interacts with that business or system by requesting or supplying services.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Actor</strong></td>
<td>[A physical component] or individual able to play one or more roles in the performance of processes. (Where non-human actors are represented as application and/or technology components, then the actors must be human.)</td>
</tr>
<tr>
<td><strong>Organisation unit</strong></td>
<td>[A physical component] or individual node in a management structure, able to fulfil one or more functions. It should have goals and objectives with measures, and a manager.</td>
</tr>
<tr>
<td><strong>Role</strong></td>
<td>[A logical component] realised or played by individual actors, specified in terms of services provided and/or processes performed and abilities required.</td>
</tr>
<tr>
<td><strong>Function</strong></td>
<td>[A logical component], a cohesive cluster of behaviors required of any organisation unit that fulfils the function, specified in terms of services provided and/or processes performed and abilities required. (It is a logical business capability, not to be confused with a managed organisation unit or discrete business service.)</td>
</tr>
<tr>
<td><strong>Business process</strong></td>
<td>[A process] that is performed by the actors in a business, with or without information technologies.</td>
</tr>
<tr>
<td><strong>Business service</strong></td>
<td>[A service] that can be requested of a business, or a component thereof. (Not to be confused with a business function.)</td>
</tr>
<tr>
<td><strong>Data entity</strong></td>
<td>[A data element] composed of data items that represent facts about a discrete business entity or event. It may be specified at a conceptual, logical or physical level. It may be mapped to data stores and/or data flows input to or output from IS services.</td>
</tr>
<tr>
<td><strong>IS (app) service</strong></td>
<td>[A service] that can be requested of a business-oriented application component, by a human actor or another application component.</td>
</tr>
<tr>
<td><strong>Application component</strong></td>
<td>[A component] of business-oriented software (e.g. CRM, Billing). It is specified logically by the IS services it provides, and sometimes also by the data entities it maintains, and/or physically as a vendor/technology specific product that can be hired, bought or built.</td>
</tr>
<tr>
<td><strong>Platform service</strong></td>
<td>[A service] that can be requested of a technology component by an application component or another technology component.</td>
</tr>
<tr>
<td><strong>Technology component</strong></td>
<td>[A component] of generic infrastructure software (e.g. OS, DBMS). It is specified logically by the platform services it provides, and/or physically as a vendor/technology specific product that can be hired or bought.</td>
</tr>
</tbody>
</table>
On nesting

- TOGAF assumes components are encapsulatable behind interfaces that expose discretely invokable services.
- Since, businesses/systems/components are nested, a service may be external to the system of interest being described, modelled or viewed at this moment, yet also internal to a wider system (viewed at some other moment).
- So concept definitions have to be written in a way that allows either or both view to be taken.

On services as behavior elements rather than structure elements

- “Service” is often defined ambiguously; people use the term for a component or an interface.
- But here, a service is a discrete behavior that can be exposed in an interface.
- So, a Web Service is an interface and a Web Service Operation is a service.

On data elements (passive structure)

- Data entities represent discrete things a business needs to remember and monitor or direct, both persistent things (e.g. a customer, a gas meter) and transient events (e.g. a payment, a meter reading).
- Each is definable as group of data items that represent facts about the business entity or event of interest.
- A data component is a structured collection of data entities accessible in one store/location.
Logical Component = Function or Interface?

A logical component is an abstract specification of an active structure element.

It may be described in terms of processes performed and abilities required, and represented using an ArchiMate Function or Role symbol.

or described in terms of services provided and represented using an ArchiMate Interface symbol.
TOGAF assigns behavior elements to logical structure elements because they are more stable than physical elements.

TOGAF assigns a “Service portfolio” to a Logical Component, which may be represented as a Function, or an Interface, in ArchiMate.

TOGAF’s Physical Technology Component embraces Systems Software (aka Platform Application), Device and Network in ArchiMate.
The phase deliverables are substantial documents.

The Architecture Requirements Specification includes the required Services.

Building blocks are recorded in catalogues, and related in matrices and diagrams.

About the two reference models, note that:

1: The III-RM is an application integration pattern, but not the only one.

A deeper dive into theory

Copyright Avancier Limited March 2016 http://avancier.website page 13 of 16

- Remember, while it helps to understand principles in this presentation, very few ever get to complete a system model.

- When you see or envisage regular behavior in the real world, you can describe it as an activity system.
- An activity system is an abstraction; its boundary and its atomic parts are determined by its stakeholders and describers.
- The atomic actors (e.g. humans in a business, “objects” in software) must each be addressable.
- The atomic activities are elementary actions deemed by describers to be performable by actors.
- It is usually assumed an atomic activity either succeeds completely or fails completely.
- Activities and longer processes have duration and throughput attributes.

- Through its boundary, systems both provide and require services. Architects start by defining provided services. They sometimes take required services for granted, assuming they will be provided by commonly available generic servers.

- Processes realises services. A principle of logic is that every atomic activity and longer-running process or value stream can be encapsulated by a service contract. So, process and service are always relatable in a 1-to-1 association.

- Structured analysis tells us that, if both structural views and behavioral views of activities are completed, then an atomic activity is an atomic process AND an atomic function. (So, to avoid duplicating associations in a meta model you might draw out atomic activity as a distinct entity, related to coarser-grained process and function entities.)
A TOGAF-compatible generic meta model

Copyright Avancier Limited March 2016 http://avancier.website page 14 of 16

System

A system or system element may be decomposed into finer-grained elements of the same type, and related to elements of other types.

Behavior element

Doings have duration and throughput attributes.

Event/Service
An event-triggered behavior that maintains system state and/or delivers value to a client.

Defined by an I/O contract that abstracts from Processes that respond to the Event or realise the Service.

Structure element

Doers are addressable.

Function/Interface
A structure that groups Services accessible to clients of the Function/Interface.

May be seen as a logical Component that abstracts from Components that actually do things.

Groups

Actor/Component
A subsystem, a unit of a system’s structure that can perform Processes and realise Services.

Replaceable by any other Actor or Component able to do the same things.

Location

Where an Interface, Actor or Component can be found.

Passive Object

An item that is made, moved, modified, used or destroyed by behavior.

Realises

Groups

TOGAF features this derived association

Performs

Orchestrates

Serves

Serves

Realises Services of

Copyright Avancier Limited March 2016 http://avancier.website
Some odd things in ArchiMate’s generic meta model

Copyright Avancier Limited March 2016 http://avancier.website page 15 of 16

For more observations, read http://grahamberrisford.com/00EAframeworks/05ArchiMate/Issues%20with%20the%20ArchiMate%20standard.htm

System

ArchiMate does not explicitly say it is for modelling discrete event driven systems

Do we use Business Object to represent a material object regardless of data? Or a material object type that a business describes in data? Or a conceptual Data Object?

Accesses could mean makes, moves, modifies, uses or destroys.

Do we use Data Object to represent a discrete Data Entity (contained in one or more Data Structures) at the Conceptual, Logical, or Physical level? What about a discrete Data Store? A Data Flow? The Data Structure contained in a Store or Flow?

Behavior element

Event-driven? Discrete steps?

Service

The definition leaves the concept open to interpretation. In practice, the symbol is abused to rename a Component in verb form and/or model what could be shown as a Function or Interface.

Active Structure element

Is it really composed of one or more Interfaces? Or does it only realise one or more Services that are exposed in one or more Interfaces?

Interface

Bound to one and only one Component. So, to model a more loosely-coupled Interface (e.g. a Web Service) you have to represent it as an Internal Active Structure Element.

Internal Active Structure Element

Is a composite of

Assigned to

Realises ▲

Serves

Assigned to

Internal Behavior Element

Function misplaced here, since it is a node in a structural decomposition, and more akin to an Interface than a Process.

Serves

Assigned to

Internal Active Structure Element

Is a composite of

Serves

A generic meta model of this kind shows generalised types of system element.

A model of a specific system also usually shows system element types rather than system element instances.

Sometimes, however, there is one and only instance of a named actor or component, so an element in a model represents one individual actor or component.
For Enterprise and Solution Architecture training and methods

http://avancier.website

ON AVANCIER TRAINING

• “Your course is brilliant I would recommend it to anyone.” Solution Architect
• “I really enjoyed your course and [have since] found the material you presented very useful.” Solutions Architect
• “explains what others (TOGAF, CMU…) leave obscure.”
• “Really good and provided me with new views on EA and SA – one of the best trainings I attended.” Enterprise Architect
• “Your exam preparation really useful (a key objective for me).” Consultant
• “One of the few people I know who can articulate this area and subject in an understandable format.” Consultant
• “It is amazing how my conversations have changed! On Monday I had three conversations that [made me] realise the positive effect the course has had... many things I now understand that I had only the smallest glimmer of before.” Infrastructure architect
• “An intensive course, covering the end-to-end of enterprise and solution architecture, from a tutor with a clear in depth knowledge and experience of his subject matter. A ‘must’ for all technologists, not just architects, it provides the structure of what all of us do on a day to day basis, business, data, systems and technology.” Solution architect
• “thank you for an interesting and stimulating course. It covered a lot of what I was expecting and more, and opened my eyes to architecture questions and concerns I need to be more familiar with... Your anecdotes and examples helped make what is often a dry subject more interesting... [it] really helped me to grab the fundamentals of what it takes to be an Enterprise Architect.” Business intelligence architect

ON AVANCIER METHODS

• “Incredibly good. I and a good chunk of the team use it a hell of a lot.” EA.
• “I stand in awe. A wealth of information and knowledge” Technical architect.
• “I have spent the last hour and a half on your website and wow!!!! It is a repository explaining so very many things MANY people ought to know and understand like clearer definitions of roles and general expectations of people who have (or claim) to be certain things."
• “Lots of good stuff here – thanks” Senior Managing Consultant
• “everything else I read on EA is mostly confused and confusing. Can’t anyone else... address the issues you raise and recognise the importance of what you’re saying?”