

Using ArchiMate with TOGAF

Part 1: Answers to nine general questions about methods

Graham Berrisford and Marc Lankhorst

Introduction

This paper is the second in a series in which we intend to shed light on the possibilities of using the ArchiMate language with an architecture method and inform you on the pitfalls you may face. This paper follows up “Using ArchiMate with an Architecture Method” [1], which outlined some of the fundamental ideas behind ArchiMate and set out nine questions to ask of a method to be used in conjunction with it.

This second paper returns to answer the nine questions we defined in the first paper (to help you integrate the use of ArchiMate with your chosen architecture method) using TOGAF as an example architecture method.

1. Does the method separate process from artifacts?
2. Does the method separate structure from behaviour?
3. Does the method separate external from internal?
4. Does the method separate logical from physical?
5. Does the method separate the general from the specific?
6. Does the method present domains as layers?
7. Does the method’s meta model align with ArchiMate’s?
8. Does the method rely on UML?
9. How abstract is the method?

This paper is the second in a series edited from conversations between Marc Lankhorst (group leader Service Architectures at [Novay](#) and a founder and teacher of ArchiMate® [2]) and Graham Berrisford (a TOGAF™ 9 [3] instructor for [Architecting-the-Enterprise](#) and author of the papers in [5]). In the following papers in this series, we will look in more detail at the conceptual differences and similarities between TOGAF and ArchiMate.

1. Does the method separate process from artifacts?

Marc: Wherever a method separates architecture process from architecture descriptions (entities, artifacts and outputs) we should be able to use the process to produce an ArchiMate-style architecture. Does TOGAF separate process from artifacts?

Graham: Inevitably, the process refers to artifacts and deliverables. And strictly speaking, TOGAF defines the ADM process steps and their outputs as “core”, meaning that if you don’t produce the outputs, then you are not following TOGAF.

Marc: Still, I have seen the ADM process used to construct an ArchiMate-style architecture. E.g. at a life insurance company here in the Netherlands.

Graham: Your phrase “ArchiMate-style architecture” is telling. You will be happy if people use something akin to the ADM process to produce ArchiMate-style architectures – if people use

ADM in a flexible way to do produce proper ArchiMate deliverables. That is one thing. Doing the reverse is a different matter altogether.

If architects use ArchiMate box and line symbols, but without the meanings you give them, and without the grammar of ArchiMate, then they aren't using the ArchiMate language. (Just as if aliens use the words in the English dictionary, but with different meanings, and in a different grammar, then they aren't speaking English.)

You won't be so happy if people use ArchiMate symbols to document another style of architecture - use words like function, service, interface, contract and component with different meanings - use ArchiMate symbols in a way contrary to the ArchiMate language.

Marc: True, although a linguist will tell you that a dictionary does not *define* the words in a language but rather *records* their meaning as they are used in practice. But of course, we want the ArchiMate language to be used in the way it was intended.

2. Does the method separate structure from behaviour?

Marc: Yes. We don't want following an architecture method to water down ArchiMate's separations between internal and external, and between structure (active & passive) and behaviour. Does the structure-behaviour distinction appear in TOGAF?

Graham: Not explicitly, but it isn't hard to find. The table below sorts a selection of TOGAF entities and artifacts into the categories recognised by ArchiMate.

Passive structure	Behaviour	Active structure offering services
Data Entity Data Component (data structure container) Contract	Service Process	Building Block (generic term) Business Function Application Technology
Data entities may be related to each in a structural data model, which TOGAF calls a class diagram (maddeningly, since classes are active structural elements).	Behavioural models include: Process Flow Diagram, Data Life Cycle Process-System Realisation Diagram (cf. UML Sequence Diagram).	Structural models include: Organisation Hierarchy, Function Hierarchy, Application Communication Diagram, Class Diagram Networked Computing Hardware Diagram.

Note however that the term building block is also used more loosely to mean any kind of architecture entity, including "process", "service" and "entity". And the term "component" has been added into the mix in TOGAF 9. And more significantly, TOGAF defines both logical and physical versions of some active structural elements.

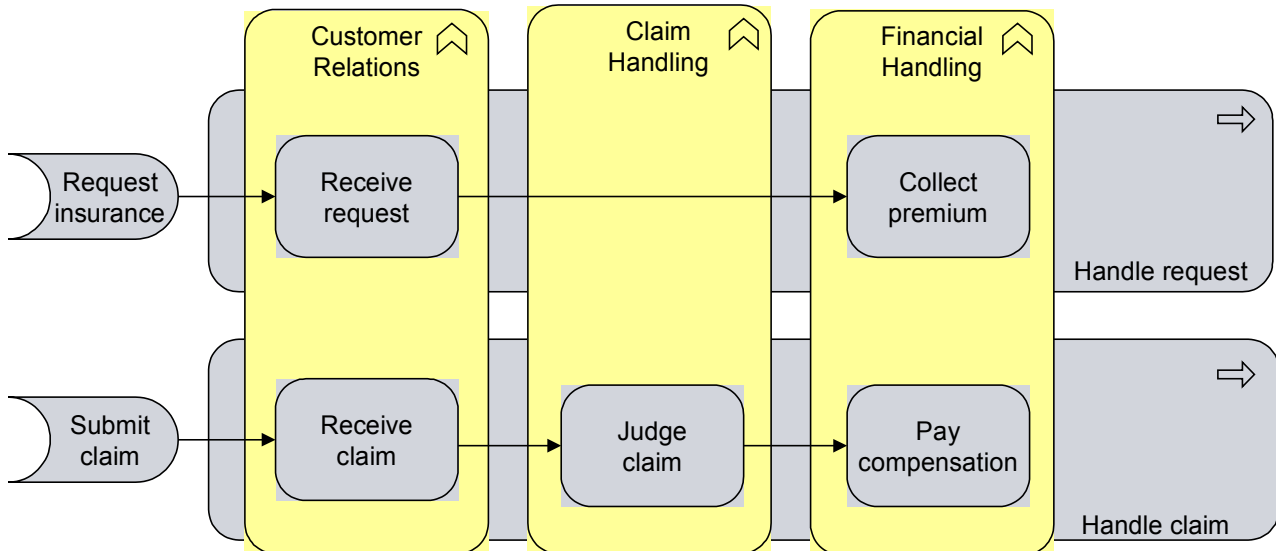
Marc: In ArchiMate, a Business Function is considered behavioural rather than structural.

Graham: Yes, ArchiMate's distinction between structure and behaviour is different from the classical one. The classical position is that behavioural and structural models are *orthogonal views of the same system specification*. This table contrasts the two views.

BEHAVIOURAL views	STRUCTURAL views
Name and connect the SUBPROCEDURES in a PROCEDURE	Name and connect the SUBSYSTEMS in a SYSTEM
E.g. a state chart a process flow chart a use case definition, an interaction or sequence diagram.	E.g. a class diagram a data flow diagram a hardware configuration diagram a data model (passive structure)
Show time-ordered control flow. Have a start and an end. Often have a main path and alternative paths.	Name subsystems and their inter-connections. Do not show time-ordered control flow.

You can decompose both system structures and process flows to several levels of granularity (though people rarely have the time and energy to manage more than two or three levels). If you complete both structural and behavioural views, then at the bottom level of decomposition, *the same elementary activities must appear in both views*.

Marc: Your description of orthogonal views relates to the contrast ArchiMate makes between processes (horizontal) and functions (vertical) in this picture.



Graham: Yes. Classical structured analysis would say a row shows a procedure composed of activities, and a column shows a subsystem into which activities are grouped according to expertise, location, or whatever. I am pretty sure UML would say a row sequences the activities in a behavioural model, and a column groups activities in a structural model.

Marc: ArchiMate classifies a business function as behavioural element, because it performs behaviour. ArchiMate has taken the subject-verb-object structure from natural language and designates the “verbs” as the behaviour.

Graham: We’ll see in the next two sections that TOGAF draws a different structural-behavioural distinction; I think the same one as in UML.

And it does seem to me that people (and TOGAF) use the word function differently in the business layer and application layer.

Kind of function	ArchiMate	TOGAF
A business function in the business layer is	A behavioural element.	A structural element: a component of a hierarchical business structure
An application function in the application layer is	A behavioural element.	A behavioural element: a process or use case.

3. Does the method separate external from internal?

Marc: ArchiMate’s external-internal dimension separates services and interfaces from their realisation or implementation by processes and components. Does the external-internal distinction appear in TOGAF?

Graham: Yes, though it isn’t the core idea that it is in ArchiMate. To begin with, TOGAF does not distinguish service from interface as ArchiMate does.

	External		Internal
TOGAF	Service	???	Component / Building Block
ArchiMate	Service	Interface	Component

In each architecture domain/layer, the TOGAF meta model distinguishes services from the components that offer them. Note however that it maps behaviour to logical components rather than physical ones.

Layer	Behavioural / External	Structural / Internal
Business	Business Service	Business Function
Application	Information System Service	Logical Application Component
Technology	Platform Service	Logical Technology Component

The service-component distinction made in the meta model is not clearly and consistently carried through the text of the ADM process, but you can find it in the two reference models.

- The Integration Information Infrastructure Reference Model (III-RM) is an SOA design pattern for applications architecture. It divides applications into three layers, each providing services to the layer above.
- The Technical Reference Model (TRM) divides the application platform interface between 'services' and 'functions'. The TRM is a hierarchical list of platform services (with a callable API) and platform functions (with no API).

Phase D uses a TRM to specify the platform services that application components require from technology components. The architect then groups or regroups the required services into service portfolios and attach them to candidate architecture building blocks.

4. Does the method separate logical from physical?

Marc: ArchiMate is concerned to distinguish external from internal. I gather TOGAF is more concerned to distinguish logical from physical.

Graham: Yes, the logical to physical distinction is very important to TOGAF. It appears in the meta model, the process (ADM) and a content classification scheme called the Enterprise Continuum.

In the meta model

Marc: I see the TOGAF meta model has a logical technology component and logical application component, but no logical business component.

Graham: Ah! That is because the logical business component is called business function! The table below shows the essence of how the logical-physical distinction appears in the TOGAF meta model.

	Logical		Physical
Layer	Behavioural / External	Structural / Internal	
Business	Business Service	Business Function	Organisation Unit
Application	Information System Service	Logical Application Component	Physical Application Component
Technology	Platform Service	Logical Technology Component	Physical Technology Component

The table shows the same three-way pattern separates concerns in each architecture domain. The bottom three rows show entities related one to another along the row. I have added titles in the rows above to show where those entities fit in the dimensions we are discussing.

Marc: Looking at the Application row in the table, I gather an information system service is realised by a logical application component, which in turn is realised by a physical application component.

Graham: Only very loosely speaking. Both those relationships are many-to-many. And your two "realised" phrases have the two different meanings we distinguished in earlier papers.

1. "Information system service is realised by a logical application component" describes the mapping from external to internal that is strongly emphasised in component-based development methods, and addressed under heading 3 above.
2. "Logical application component is realised by a physical application component" describes the mapping from vendor-neutral (logical) to vendor-specific (physical) specifications that is strongly emphasised in public sector methods and frameworks.

Marc: TOGAF appears to equate "internal" with "structural" and "external" with "behavioural",

Graham: No, it only looks like that in the tables drawn here. Behaviour embraces both external services and internal processes. And remember I am giving you my best interpretation of what 300 or more TOGAF authors meant to say or suggest, collectively, after their contributions have been edited by other authors over a decade or more! What I present here is an attempt to impose a systematic view on the TOGAF text.

In the process (ADM)

The logical-physical distinction appears in the progress of the ADM process thus:

	Logical	Physical	
ADM	Phases A, B, C, D	Phase E	Phase G
Role	Enterprise Architects		Solution Architects
Task	Define and complete	Define potential, sketchy	Complete, develop and deploy
Products	Architecture Building Blocks Services and Logical Components	Solution Building Blocks Physical Components	

Architecture building blocks are logical. Solution building blocks are *more* physical, but "may still be considerably abstracted from solution architecture, design, or implementation views". (When TOGAF practitioners get more physical and/or more detailed in phases A to D, then that suggests they are using ADM for solution architecture rather than enterprise architecture.)

In a content classification

The logical-physical distinction is between higher and lower levels in the content classification scheme known as the Enterprise Continuum, discussed below.

Aside: from ArchiMate language to Architecture Method

ArchiMate is not without implications for an Architecture Method. The table below shows some correspondences between transformations in ArchiMate, and transformations in TOGAF. But notice that the two realisation processes (sections 3 and 4 above) are different.

Requirement-to-solution process threads	ArchiMate	TOGAF
Realisation (1)	External to Internal	
Realisation (2)		Logical to Physical
Construction	Behaviour to Structure	Behaviour to Structure
Business to IT	Business, Apps, Infrastructure	Business, Data, Apps, Technology
Specialisation		General to Specific
Migration		Baseline to Target

5. Does the method separate the specific from the general?

Graham: Enterprise architecture is cross-organisational and strategic. It is about generalities: general principles, general standards, industry reference models, common systems, design patterns, reusable components. Enterprise architects deal largely in generalisations, be they at the logical or physical level.

Marc: Generalisation is not a big deal in the ArchiMate philosophy, though you can model it of course using the generalisation relationship. Does the general-to-specific dimension appear in TOGAF?

Graham: Yes. It appears as the horizontal axis in the two-dimensional content classification scheme called the Enterprise Continuum, which I can represent in a table thus:

The Enterprise Continuum		Universal	Fairly generic	Fairly specific	Uniquely configured
		Foundation	Common System	Industry	Organisation
Ideal	Req'ment				
Logical	Architecture Continuum	Foundation Architecture	Common System Architecture	Industry Architecture	Organisation Architecture
Physical	Solutions Continuum	Products and Services *	Common System Solutions	Industry Solutions	Organisation Solutions
Real	Deployed Solutions				

This schema is designed to use as an index to the building blocks and reference models in an architecture repository.

* Note that the naming of this cell is odd, since vendor-specific products and services can appear in any cell in the lower half of this table.

6. Does the method present domains as layers?

Marc: ArchiMate presents the three architecture domains (business, application and technology) as layers. Does TOGAF present the same idea, that architecture domains are layers? In the ADM "crop circles", we see approximately the same separation in three domains.

Graham: You could say the idea is implicit. There is one over-complex nested box diagram that could be read as making the idea explicit. But the idea is not thoroughly worked through. To do this would require also a consistent working through of the external-internal distinction (ArchiMate speak) or service-building block distinction (TOGAF speak).

7. Does method's meta model align with ArchiMate's?

Marc: Does the TOGAF meta model align with ArchiMate's?

Graham: Hmm... You cannot fully understand TOGAF by studying its explicit meta model. As a new arrival in TOGAF 9 (derived from the meta model in SAP's EAF) it does not yet fully match the implicit meta model in the TOGAF text.

A presentation on "Architecture meta models" that compares these two meta models with others, and introduces the challenges we are addressing here, can be found in [5].

Marc: The Open Group Architecture Forum has to address its own challenges. Some politics are reflected in TOGAF's internal inconsistencies. Convergence of explicit and implicit meta models is a challenge the Architecture Forum needs to resolve, ideally before convergence with ArchiMate. However, the explicit TOGAF meta model doesn't look too far removed from ArchiMate's.

Graham: I have listed some initial mappings in the table below. We should review each one in detail.

First-cut entity mapping table (needing further research)	
ArchiMate Entities	TOGAF Entities
Generic (meta meta) level	
Service	Service
Interface	Contract
Behaviour element	Function (Process)
Structure element	Function (Component)
Business architecture	
Actor	Actor Organisation unit
Role	Role
Business Function	Function Capability
Product Business Object	Product
Contract	Contract
Business Service	Business Service
Business Interface	
Data architecture	
Data Object	Data Entity
N/A	Logical Data Component
Artifact	Physical Data Component
Application architecture	
Application Component	Application Component
N/A	Logical Application Component
Artifact	Physical Application Component
Application Service Application Interface	Information System Service
Infrastructure or Technology architecture	
Node	Technology Component
N/A	Logical Technology Component
Device System Software	Physical Technology Component
Infrastructure Service Infrastructure Interface	Platform Service

Graham: The table is superficial. There are two challenges here:

- First, the table compares only meta models. It relies on the new TOGAF meta model definitions. But to the TOGAF practitioner, it is not always clear how ADM artifacts and outputs are constructed using the entities in this new meta model.
- Second, some of the mappings are questionable. Mappings that look obvious at first glance may disintegrate on closer inspection. We need to do a thorough entity-by-entity comparison, comparing the entities' definitions, attributes and relationships. And at the same time considering effect of recursive decomposition on the entity definitions.

We will present a more detailed analysis – comparing entities in the two meta models –in a later paper.

8. Does the method rely on UML?

Marc: ArchiMate is not intended as a competitor to UML for software architecture. Does TOGAF use UML?

Graham: No. TOGAF is concerned with the application portfolio rather than application design. Class diagrams are mentioned in different context from software design. Interaction diagrams are not mentioned (where they could be) in the definition of a process-system realisation diagram.

9. How abstract is the method?

Graham: Enterprise architecture is cross-organisational and strategic. Enterprise architecture diagrams are usually more abstract than the solution architecture diagrams.

Marc: How abstract is TOGAF?

Graham: Very. It tells us “physical elements in an enterprise architecture may still be considerably abstracted from solution architecture, design, or implementation views”.

Conclusions

In the answers to the questions of the previous sections, it has become clear that ArchiMate and TOGAF share some important concepts but also differ in several aspects. Our first impression is that the two are usable together, but that we have to be very clear on these discrepancies. Later instalments in this series will have a deeper look at the fundamentals of both and investigate the mapping between their meta model entities in more detail.

References

- [1] Graham Berrisford & Marc Lankhorst, “Using ArchiMate with an Architecture Method”, *Via Nova Architectura*, June 2009. <http://www.via-nova-architectura.org/magazine/magazine/using-archimate-with-an-architecture-method.html>
- [2] The Open Group, *ArchiMate® 1.0 Specification*. Van Haren Publishing, 2009. <http://www.opengroup.org/archimate>, <http://www.opengroup.org/bookstore/catalog/c091.htm>,
- [3] The Open Group, *TOGAF™ Version 9*. Van Haren Publishing, 2009. <http://www.opengroup.org/togaf>
- [4] Marc Lankhorst, Hans van Drunen, “Enterprise Architecture Development and Modelling – Combining TOGAF and ArchiMate”, *Via Nova Architectura*, 21 March 2007. <http://www.via-nova-architectura.org/magazine/magazine/enterprise-architecture-development-and-mode.html>
- [5] Graham Berrisford, Papers accessible at <http://avancier.co.uk>, in particular:

“What is a Function?”
“Architecture Meta Models”
Papers on “Abstraction” in the “Library”.

Copyright

This paper was edited from conversations between Marc Lankhorst (group leader Service Architectures at [Novay](#) and a founder and teacher of ArchiMate® [2]) and Graham Berrisford (a TOGAF™ 9 [3] instructor for [Architecting-the-Enterprise](#) and author of the papers in [5]).

Creative Commons Attribution-No Derivative Works Licence 2.0

No Derivative Works: You may copy, distribute, display only complete and verbatim copies of this document, not derivative works based upon it.

Attribution: You may copy, distribute and display this copyrighted work only if you clearly credit the authors Graham Berrisford of [Avancier Limited](#) and Marc Lankhorst of [Novay](#) before the start and include this footnote at the end.

For more information about the licence, see <http://creativecommons.org>

ArchiMate® and TOGAF™ are registered trademarks of The Open Group.

Graham Berrisford
graham.berrisford@architecting-the-enterprise.com

Marc Lankhorst
marc.lankhorst@novay.nl