THE OPEN GROUP SERIES

# ArchiMate® 2.0 Specification

Van Haren PUBLISHING

THE Open GROUP

Open Group Standard

# ArchiMate® 2.0 Specification

## The Open Group Publications available from Van Haren Publishing

**The TOGAF Series:**
TOGAF® Version 9.1
TOGAF® Version 9.1 - A Pocket Guide
TOGAF® 9 Foundation Study Guide, 2nd Edition
TOGAF® 9 Certified Study Guide, 2nd Edition

**The Open Group Series:**
Cloud Computing for Business – The Open Group Guide
ArchiMate 2.0 - A Pocket Guide
ArchiMate® 2.0 Specification

**The Open Group Security Series:**
Open Information Security Management Maturity Model (O-ISM3)
Open Enterprise Security Architecture (O-ESA)
Risk Management - The Open Group Guide

All titles are available to purchase from:
www.opengroup.org
www.vanharen.net
and also many international and online distributors.

Open Group Standard

# ArchiMate® 2.0 Specification


Title: **ArchiMate® 2.0 Specification**

A Publication of: The Open Group

Publisher: Van Haren Publishing, Zaltbommel, www.vanharen.net

ISBN Hard copy: 978 90 8753 692 3
ISBN eBook: 978 90 8753 947 4

Edition: First edition, second impression, June 2012

Design and Layout: CO2 Premedia, Amersfoort-NL



Document Number: C118

Comments relating to the material contained in this document may be submitted to:
The Open Group
Apex Plaza
Forbury Road
Reading
Berkshire, RG1 1AX
United Kingdom
or by electronic mail to: ogspecs@opengroup.org

# Contents

# Table of Figures

# Preface

## The Open Group

The Open Group is a global consortium that enables the achievement of business objectives through IT standards. With more than 400 member organizations, The Open Group has a diverse membership that spans all sectors of the IT community – customers, systems and solutions suppliers, tool vendors, integrators, and consultants, as well as academics and researchers – to:

- Capture, understand, and address current and emerging requirements, and establish policies and share best practices
- Facilitate interoperability, develop consensus, and evolve and integrate specifications and open source technologies
- Offer a comprehensive set of services to enhance the operational efficiency of consortia
- Operate the industry's premier certification service

Further information on The Open Group is available at www.opengroup.org.

The Open Group publishes a wide range of technical documentation, most of which is focused on development of Open Group Standards and Guides, but which also includes white papers, technical studies, certification and testing documentation, and business titles. Full details and a catalog are available at www.opengroup.org/bookstore.

Readers should note that updates – in the form of Corrigenda – may apply to any publication. This information is published at www.opengroup.org/corrigenda.

## This Document

This document is The Open Group Standard for the ArchiMate 2.0 Specification.

Issue 2.0 includes a number of corrections, clarifications, and improvements compared to the previous issue, as well as two optional language extensions: the Motivation extension and the Implementation and Migration extension.

## Intended Audience

The intended audience of this Technical Standard is threefold:

- Enterprise architecture practitioners, such as architects (application, information, process, infrastructure, products/services, and, obviously, enterprise architects), senior and operational management, project leaders, and anyone committed to work within the reference framework defined by the enterprise architecture. It is assumed that the reader has a certain skill level and is effectively committed to enterprise architecture. Such a person is most likely the architect – that is, someone who has affinity with modeling techniques, knows his way around the organization, and is familiar with information technology.
- Those who intend to implement ArchiMate in a software tool. They will find a complete and detailed description of the language in this document.
- The academic community, on which we rely for amending and improving the language based on state-of-the-art research results in the architecture field.

## Structure

The structure of this Technical Standard is as follows:

- Chapter 1, Introduction, provides a brief introduction to the purpose of this standard.
- Chapter 2, Language Structure, presents some general ideas, principles, and assumptions underlying the development of the ArchiMate metamodel and introduces the ArchiMate framework.
- Chapter 3, Business Layer, covers the definition and usage of the business layer concept, together with examples.
- Chapter 4, Application Layer, covers the definition and usage of the application layer concept, together with examples.
- Chapter 5, Technology Layer, covers the definition and usage of the technical infrastructure layer concept, together with examples.
- Chapter 6, Cross-Layer Dependencies, and Chapter 7, Relationships, cover the definition of relationship concepts in a similar way.
- Chapter 8, Architecture Viewpoints, presents and clarifies a set of architecture viewpoints, developed in ArchiMate based on practical experience. All ArchiMate viewpoints are described in detail. For each viewpoint the comprised concepts and relationships, the guidelines for the viewpoint use, and the goal and target group and of the viewpoint

are specified. Furthermore, each viewpoint description contains example models.

- Chapter 9, Language Extension Mechanisms, handles extending and/or specializing the ArchiMate language for specialized or domain-specific purposes.
- Chapter 10, Motivation Extension, describes an optional language extension with concepts, relationships, and viewpoints for expressing the motivation for an architecture (e.g., stakeholders, concerns, goals, principles, and requirements).
- Chapter 11, Implementation and Migration Extension, describes an optional language extension with concepts, relationships, and viewpoints for expressing the implementation and migration aspects of an architecture (e.g., project, programs, plateaus, and gaps).
- Chapter 12, Future Directions, is an informative chapter that identifies extensions and directions for developments in the next versions of the language.

# Trademarks

Boundaryless Information Flow™ is a trademark and ArchiMate®, Jericho Forum®, Making Standards Work®, Motif®, OSF/1®, The Open Group®, TOGAF®, UNIX®, and the "X" device are registered trademarks of The Open Group in the United States and other countries.

Java® is a registered trademark of Oracle and/or its affiliates.

MDA®, Model Driven Architecture®, OMG®, and UML® are registered trademarks and BPMN™, Business Process Modeling Notation™, MOF™, and Unified Modeling Language™ are trademarks of the Object Management Group..

All other brands, company, and product names are used for identification purposes only and may be trademarks that are the sole property of their respective owners.

# Acknowledgements

The Open Group gratefully acknowledges the contribution of the following people in the development of this Open Group Standard:

- Maria-Eugenia Iacob, University of Twente
- Henk Jonkers, BiZZdesign BV
- Marc M. Lankhorst, Novay
- Erik (H.A.) Proper, Public Research Centre Henri Tudor & Radboud University Nijmegen
- Dick A.C. Quartel, BiZZdesign BV

The Open Group and ArchiMate project team would like to thank in particular the following individuals for their support and review of this Open Group Standard:

- Iver Band, Standard Insurance Company
- Mary Beijleveld, UWV
- Alexander Bielowski, Software AG
- Adrian Campbell, Ingenia Consulting
- John Coleshaw, QA Ltd.
- Jörgen Dahlberg, Biner Consulting
- Garry Doherty, The Open Group
- Wilco Engelsman, BiZZdesign BV
- Roland Ettema, Logica
- Henry M. Franken, BiZZdesign BV
- Kirk Hansen, Kirk Hansen Consulting
- Jos van Hillegersberg, University of Twente
- Andrew Josey, The Open Group
- Louw Labuschagne, Real IRM
- Veer Muchandi, Hewlett-Packard
- Bill Poole, JourneyOne
- Henk Volbeda, Sogeti
- Egon Willemsz, UWV

The results presented in this Open Group Standard have largely been produced during the ArchiMate project, and The Open Group gratefully acknowledges the contribution of the many people – former members of the project team – who have contributed to them.

The ArchiMate project comprised the following organizations:

- ABN AMRO
- Centrum voor Wiskunde en Informatica
- Dutch Tax and Customs Administration
- Leiden Institute of Advanced Computer Science
- Ordina
- Radboud Universiteit Nijmegen
- Stichting Pensioenfonds ABP
- Novay

# Referenced Documents

The following documents are referenced in this Open Group Standard:

[1] ISO/IEC 42010:2007, Systems and Software Engineering – Recommended Practice for Architectural Description of Software-Intensive Systems, Edition 1.

[2] Enterprise Architecture at Work: Modeling, Communication, and Analysis, M.M. Lankhorst et al, Springer, 2005.

[3] Architecture Principles: The Cornerstones of Enterprise Architecture, D. Greefhorst, E. Proper, Springer, 2011.

[4] The Open Group Architecture Framework TOGAF, Version 9, 2009.

[5] A Framework for Information Systems Architecture, J.A. Zachman, IBM Systems Journal, Volume 26, No. 3, pp. 276–292, 1987.

[6] ITU Recommendation X.901 | ISO/IEC 10746-1:1998, Information Technology – Open Distributed Processing – Reference Model – Part 1: Overview, International Telecommunication Union, 1996.

[7] Unified Modeling Language: Infrastructure, Version 2.0 (formal/05-05-05), Object Management Group, March 2006.

[8] Extending and Formalizing the Framework for Information Systems Architecture, J.F. Sowa, J.A. Zachman,, IBM Systems Journal, Volume 31, No. 3, pp. 590-616, 1992.

[9] Enterprise Ontology: Theory and Methodology, J.L.G. Dietz, Springer, 2006.

[10] Unified Modeling Language: Superstructure, Version 2.0 (formal/05-07-04), Object Management Group, August 2005.

[11] A Business Process Design Language, H. Eertink, W. Janssen, P. Oude Luttighuis, W. Teeuw, C. Vissers, in Proceedings of the First World Congress on Formal Methods, Toulouse, France, September 1999.

[12] Enterprise Business Architecture: The Formal Link between Strategy and Results, R. Whittle, C.B. Myrick, CRC Press, 2004.

[13] Composition of Relations in Enterprise Architecture, R.v. Buuren, H. Jonkers, M.E. Iacob, P. Strating, in Proceedings of the Second International Conference on Graph Transformation, pp. 39–53, Edited by H. Ehrig et al, Rome, Italy, 2004.

[14] Viewpoints: A Framework for Integrating Multiple Perspectives in System Development, A. Finkelstein, J. Kramer, B. Nuseibeh, L. Finkelstein, M. Goedicke, in International Journal on Software

Engineering and Knowledge Engineering, Volume 2, No. 1, pp. 31–58, 1992.

[15] Viewpoints for Requirements Definition, G. Kotonya, I. Sommerville, IEE/BCS Software Engineering Journal, Volume 7, No. 6, pp. 375–387, November 1992.

[16] Paradigm Shift – The New Promise of Information Technology, D. Tapscott, A. Caston, New York: McGraw-Hill, 1993.

[17] The 4+1 View Model of Architecture, P.B. Kruchten, IEEE Software, Volume 12, No. 6, pp. 42–50, 1995.

[18] Model-Driven Architecture: Applying MDA to Enterprise Computing, D. Frankel, Wiley, 2003.

[19] Performance and Cost Analysis of Service-Oriented Enterprise Architectures, H. Jonkers, M. E. Iacob, in Global Implications of Modern Enterprise Information Systems: Technologies and Applications, Edited by A. Gunasekaran, IGI Global, 2009.

[20] Business Process Modeling Notation Specification (dtc/06-02-01), Object Management Group, February 2006.

[21] The Chaos Report, The Standish Group, 1994.

[22] No Silver Bullet: Essence and Accidents of Software Engineering, F.P. Brooks, IEEE Computer, 20(4):10–19, 1987.

[23] Managing Successful Programs, Office of Government Commerce (OGC), Stationery Office Books, 2007.

[24] Managing Successful Projects with PRINCE2 – 2009 Edition, Office of Government Commerce (OGC), Stationery Office Books, 2009.

[25] A Guide to the Project Management Body of Knowledge (PMBoK Guide), Fourth Edition, Project Management Institute, 2009.

Chapter 1

# Introduction

An architecture is typically developed because key people have concerns that need to be addressed by the business and IT systems within the organization. Such people are commonly referred to as the "stakeholders" in the system. The role of the architect is to address these concerns, by identifying and refining the requirements that the stakeholders have, developing views of the architecture that show how the concerns and the requirements are going to be addressed, and by showing the trade-offs that are going to be made in reconciling the potentially conflicting concerns of different stakeholders. Without the architecture, it is unlikely that all the concerns and requirements will be considered and met.

Architecture descriptions are formal descriptions of an information system, organized in a way that supports reasoning about the structural and behavioral properties of the system and its evolution. They define the components or building blocks that make up the overall information system, and provide a plan from which products can be procured, and subsystems developed, that will work together to implement the overall system. It thus enables you to manage your overall IT investment in a way that meets the needs of your business.

To provide a uniform representation for diagrams that describe enterprise architectures, the ArchiMate enterprise architecture modeling language has been developed. It offers an integrated architectural approach that describes and visualizes the different architecture domains and their underlying relations and dependencies.

ArchiMate is a lightweight and scalable language in several respects:

- Its architecture framework is simple but comprehensive enough to provide a good structuring mechanism for architecture domains, layers, and aspects.
- The language incorporates the concepts of the "service orientation" paradigm that promotes a new organizing principle in terms of (business,

application, and infrastructure) services for organizations, with far-reaching consequences for their enterprise architecture.

The role of the ArchiMate standard is to provide a graphical language for the representation of enterprise architectures over time (i.e., including transformation and migration planning), as well as their motivation and rationale. The evolution of the standard is closely linked to the developments of the TOGAF standard and the emerging results from The Open Group forums and work groups active in this area. As a consequence, the ArchiMate standard does not provide its own set of defined terms, but rather follows those provided by the TOGAF standard.

This is Issue 2.0 of the Technical Standard, which contains a number of corrections, improvements, and clarifications in the description of the core language as described in Issue 1.0, as well as two optional extensions of the language: the Motivation extension and the Implementation and Migration extension.

This specification contains the formal definition of ArchiMate as a visual design language with adequate concepts for specifying inter-related architectures, and specific viewpoints for selected stakeholders. This is complemented by some considerations regarding language extension mechanisms, analysis, and methodological support. Furthermore, this document is accompanied by a separate document, in which certification and governance procedures surrounding the specification are specified.

Chapter 2

# Language Structure

The unambiguous specification and description of enterprise architecture's components and especially of their relationships requires an architecture modeling language that addresses the issue of consistent alignment and facilitates a coherent modeling of enterprise architectures.

This chapter presents the construction of the ArchiMate architecture modeling language. The precise definition and illustration of its generic set of core concepts and relationships follow in Chapters 3, 4, 5, 6 and 7. The concepts and relationships of the two language extensions are described in more detail in Chapters 10 and 11. They provide a proper basis for visualization, analysis, tooling, and use of these concepts and relationships.

Sections 2.1 through 2.5 discuss some general ideas, principles, and assumptions underlying the development of the ArchiMate metamodel. Section 2.6 presents the ArchiMate framework, which is used in the remainder of this document as a reference taxonomy scheme for architecture concepts, models, viewpoints, and views. Sections 2.7 and 2.8 describe the basic structure of the two language extensions. Section 2.9 briefly describes the relationship between ArchiMate and TOGAF.

## 2.1 Design Approach

A key challenge in the development of a general metamodel for enterprise architecture is to strike a balance between the specificity of languages for individual architecture domains, and a very general set of architecture concepts, which reflects a view of systems as a mere set of inter-related entities. Figure 1 illustrates that concepts can be described at different levels of specialization.

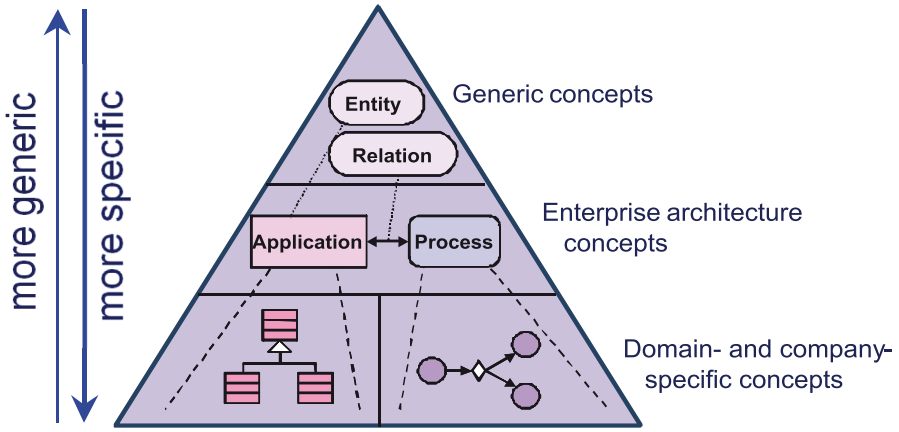


Figure 1: Metamodels at Different Levels of Specificity

At the base of the triangle we find the metamodels of the architecture modeling concepts used by specific organizations, as well as a variety of existing modeling languages and standards; UML is an example of a language in this category. At the top of the triangle we find the "most general" metamodel for system architectures, essentially a metamodel that merely comprises notions such as "entity" and "relation".

The design of the ArchiMate language started from a set of relatively generic concepts (higher up in the pyramid). These have been specialized towards application at different architectural layers, as explained below in the following sections.

The most important design restriction on the language is that it has been explicitly designed to be as small as possible, but still usable for most enterprise architecture modeling tasks. Many other languages, such as UML 2.0, try to accommodate all needs of all possible users. In the interest of simplicity of learning and use, ArchiMate has been limited to the concepts that suffice for modeling the proverbial 80% of practical cases.

## 2.2 Core Concepts

The core language consists of three main types of elements (note, however, that the model elements often represent *classes* of entities in the real world): *active structure* elements, *behavior* elements, and *passive structure* elements

(*objects*). The active structure elements are the business actors, application components, and devices that display actual behavior; i.e., the 'subjects' of activity (right side of the Figure 2).

> An active structure element is defined as an entity that is capable of performing behavior.

Then there is the behavioral or dynamic aspect (center of Figure 2). The active structure concepts are assigned to behavioral concepts, to show who or what performs the behavior.

> A behavior element is defined as a unit of activity performed by one or more active structure elements.

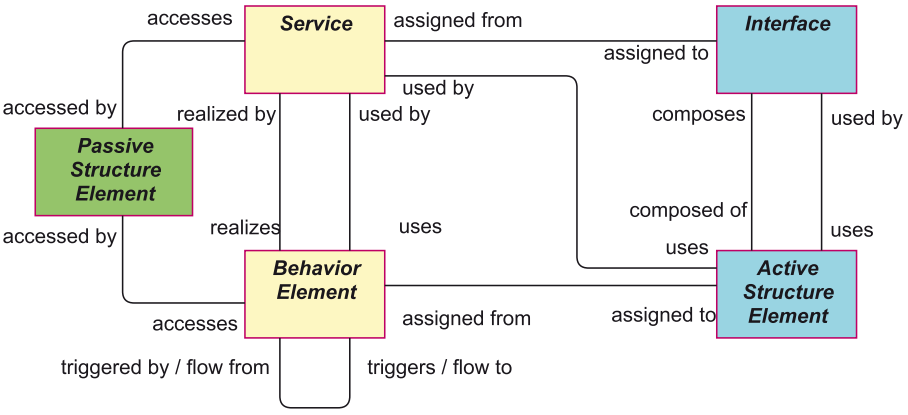


Figure 2: Generic Metamodel: The Core Concepts of ArchiMate[1]

The passive structure elements are the objects on which behavior is performed.

1 In this figure, and all the other metamodel pictures in this document, a convention for role names of relationships is used that is similar to UML (but using verbs instead of nouns). For example, a Behavior Element *realizes* a Service, and a Service *is realized by* a Behavior Element. If no cardinality is shown for a relationship end, a default of 0..* (zero or more) is assumed; if the default does not apply, the cardinality is shown explicitly in the metamodel.

> A passive structure element is defined as an object on which behavior is performed.

In the domain of information-intensive organizations, which is the main focus of the language, passive structure elements are usually information or data objects, but they may also be used to represent physical objects. These three aspects – active structure, behavior, and passive structure – have been inspired by natural language, where a sentence has a subject (active structure), a verb (behavior), and an object (passive structure).

Second, we make a distinction between an external view and an internal view on systems. When looking at the behavioral aspect, these views reflect the principles of service orientation.

> A service is defined as a unit of functionality that a system exposes to its environment, while hiding internal operations, which provides a certain value (monetary or otherwise).

Thus, the service is the externally visible behavior of the providing system, from the perspective of systems that use that service; the environment consists of everything outside this providing system. The value provides the motivation for the service's existence. For the external users, only this exposed functionality and value, together with non-functional aspects such as the quality of service, costs, etc., are relevant. These can be specified in a contract or Service Level Agreement (SLA). Services are accessible through interfaces, which constitute the external view on the active structural aspect.

> An interface is defined as a point of access where one or more services are made available to the environment.

An interface provides an external view on the service provider and hides its internal structure.