ENTERPRISE ARCHITECT

User Guide Series

ArchiMate Modeling Language

Author: Sparx Systems
Date: 10/11/2023
Version: 16.1

CREATED WITH ENTERPRISE ARCHITECT

Table of Contents

ArchiMate Modeling Language 4
- Getting Started 6
- Introduction to ArchiMate 7
- The ArchiMate Language 10
  - Language Definitions 12
  - Language Structure 21
  - Generic Metamodel 22
  - Elements 26
    - Changing Element Presentation 27
    - Motivation Elements 31
      - Motivation Example Diagrams 33
    - Strategy Layer Elements 36
      - Strategy Layer Example Diagrams 37
    - Business Layer Elements 38
      - Business Layer Example Diagrams 41
    - Application Layer Elements 42
      - Application Layer Example Diagrams 44
    - Technology Layer Elements 45
      - Technology Examples 47
    - Table of Physical Layer Elements 48
      - Physical Examples 49
    - Table of Implementation and Migration Layer Elements 50
      - Implementation and Migration Example 51
    - Composite Elements 52
  - Relationships 53
    - Relationships Overview 55
    - Structural Relationships 59
    - Dependency Relationships 60
    - Dynamic Relationships 61
    - Other Relationships 62
    - Relationship Connectors 63
  - Alignment between Layers 64
- Customizing the ArchiMate Language 66
  - Example Profile 69
- Architecture Views and Viewpoints 70
  - Overview of Views and Viewpoints 71
  - Basic Viewpoints 75
    - Organization Viewpoint 76
    - Application Structure Viewpoint 77
    - Information Structure Viewpoint 78
    - Technology Viewpoint 79
    - Layered Viewpoint 80
    - Physical Viewpoint 82
    - Product Viewpoint 83
    - Application Usage Viewpoint 84
    - Technology Usage Viewpoint 85
    - Business Process Cooperation Viewpoint 86

Application Cooperation Viewpoint 87
Service Realization Viewpoint 88
Implementation and Deployment Viewpoint 89
Motivation Viewpoints 90
Stakeholder Viewpoint 91
Goal Realization Viewpoint 92
Requirements Realization Viewpoint 93
Motivation Viewpoint 94
Strategy Viewpoints 95
Strategy Viewpoint 96
Capability Map Viewpoint 97
Value Stream Viewpoint 98
Outcome Realization Viewpoint 99
Resource Map Viewpoint 100
Implementation and Migration Viewpoints 101
Project Viewpoint 102
Migration Viewpoint 103
Implementation and Migration Viewpoint 104
Custom ArchiMate Viewpoints 105
Using ArchiMate with Enterprise Architecture 108
Exchanging ArchiMate Models 113
Generate Model Exchange File 114
Import ArchiMate Model Exchange File 116
Migrate to the Latest ArchiMate Version 118

ArchiMate Modeling Language

Create and Visualize Enterprise Architecture Models, Views and Viewpoints

ArchiMate® is one of the Enterprise Architecture tools integrated with Enterprise Architect.

ArchiMate® is an open-standard enterprise architecture language managed by The Open Group, based on the IEEE 42010 standard. It is "a visual language with a set of default iconography for describing, analyzing, and communicating many concerns of Enterprise Architectures as they change over time." It is used by a variety of modelers and stakeholders; Enterprise Architects will create and maintain models but a range of other stakeholders will view the models and representations of them. These include Solution Architects, Business Analysts, Executives and Business Line Managers, just to mention a few. The language is used to represent elements from all layers and aspects of an organization, from drivers and stakeholders to devices and networks.

Overview of Topics

This table lists the main topics that describe the implementation of the ArchiMate language in Enterprise Architect. The architect will learn how to start modeling with ArchiMate and will be introduced to the features available within the tool to create expressive diagrams and views with ArchiMate. Further topics will describe the creation of Views and Viewpoints and how to create diagrams that contain elements and relationships and will detail all these concepts. Later topics discuss the migration from earlier versions of the language to the latest version and also how to exchange models.

Brief Introduction	This topic provides an introduction to ArchiMate in Enterprise Architect and discusses its relevance and importance in creating and maintaining Enterprise Architecture models.
Getting Started	This topic provides the information needed to immediately start modeling with ArchiMate, including the ArchiMate Perspective and the Model Wizard patterns.
Using ArchiMate	This topic covers the basic tool features to create Packages, diagrams and elements, including showing how existing and new elements can be added to diagrams.
Views and Viewpoints	This topic describes the archimate views and viewpoints giving examples of each viewpoint available for immediate creation from the model wizard pattern tool.
ArchiMate Language	This topic describes the ArchiMate Language, including the language definitions, structure and metamodel, and the concepts, divided into elements and relationships.
Customising the Language	This topic describes the principles and mechanisms for extending the language to add industry- or organization-specific information in the form of properties, to both

	elements and connectors.
Exchanging Models	This topic describes the exchange of ArchiMate models created in Enterprise Architect with other tools, enabling models to be both imported and exported using the xml-based exchange format.
Version Migration	This topic describes how to upgrade models containing elements, relationships and diagrams from one version of ArchiMate to a later version.

Getting Started

Welcome to the ArchiMate language, fully integrated with Enterprise Architect. The tool not only supports all of the language features, including productivity features, but also provides a collaboration platform that will allow architects and others from management down to development and support teams to view and contribute to the architecture and the value it brings to the organization.

About ArchiMate

ArchiMate is a technical standard managed by *The Open Group;* it is partly inspired by and based on the concepts of the IEEE 42010 standard (formerly IEEE-1471). Work began on the standard as early as 2002 in the Netherlands by a project team from the *Telematica Instituut* in association with a number of Dutch partners from government, industry and academia. It has evolved significantly and has become a general purpose standard that is used for analyzing, specifying and documenting architectures at any level from strategy to deployment.

Enterprise Architect has supported modeling in ArchiMate for a number of years; the current version of Enterprise Architect supports ArchiMate 3.1 and all its syntax, semantics and viewpoints.

Objectives

"The standard is the specification of the ArchiMate Enterprise Architecture modeling language, a visual language with a set of default iconography for describing, analyzing, and communicating many concerns of Enterprise Architectures as they change over time. The standard provides a set of entities and relationships with their corresponding iconography for the representation of Architecture Descriptions." *ArchiMate ® 3.1 Specification p.1*

Value

The language has been developed with a recognition of the varied stakeholder groups, both technical and non-technical, that need to interact with the architecture. It has provided a large number of features that support this varied group and their need to communicate and collaborate, including:

- A uniform representation for diagrams that describe Enterprise Architectures
- Views and Viewpoints to address stakeholder specific concerns
- Mechanisms to visualize interrelated architectures including different domains
- Syntactic devices for structuring mechanism for architecture domains, layers, and aspects
- Visual devices that allow language elements to be presented with tailored notations for stakeholder groups
- Support for conceptualization by relation abstract and concrete concepts using realization
- Service-orientation to distinguish and relate the Business, Application, and Technology Layers of Enterprise Architectures

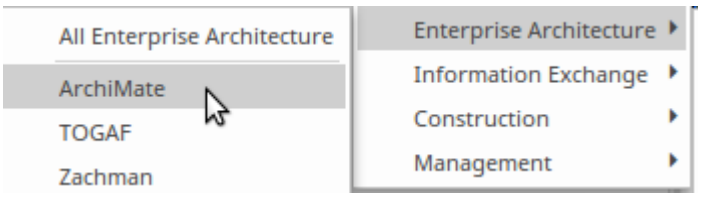
Introduction to ArchiMate

Enterprise Architect supports a wide range of languages and frameworks, which are all available within the tool depending on the edition that you are running. This provides great flexibility as it allows languages to be used in combination; for example, even though most of your architecture might be developed using ArchiMate there could be occasions when you need to create a Mind Map to document a stakeholder workshop or run a simulation of a Business Process to fully understand it. The tool provides a series of Perspectives that let you select a single aspect, language or discipline within the tool, allowing you to focus, but when you need to you can simply switch to an alternative perspective.

ArchiMate Perspective

The ArchiMate Perspective allows you to focus on architecture analysis and modeling using the ArchiMate language. To switch to the ArchiMate Perspective you need to:

1. Click on the <perspective name> option in the top right corner of the application title bar.
2. Select *Enterprise Architecture* > *ArchiMate* from the drop down



Enterprise Architect now will allow you to focus on modeling with ArchiMate and will limit the available diagrams and toolbox and language features to the ArchiMate technology. It will also open the Start Page 'Create from Pattern' tab (Model Wizard), allowing you to kick-start your modeling with a series of pre-built patterns that fully support the ArchiMate Viewpoint mechanism.

Please note: If you select the 'ArchiMate' perspective, you will be using ArchiMate 3. If you want to use ArchiMate 1 or 2, select the perspective 'All Enterprise Architecture' or create your own custom perspective.

Model Wizard Patterns

The Model Wizard is a productivity tool that allows you to create compliant model content based on the chosen example viewpoint, selected from these viewpoint groups:

- Basic Viewpoints
- Motivation Viewpoints
- Strategy Viewpoints
- Implementation and Migration Viewpoints

The patterns contained in the Model Wizard are fully documented and provide valuable information that describes the viewpoint and how it can be used.

Basic Viewpoints

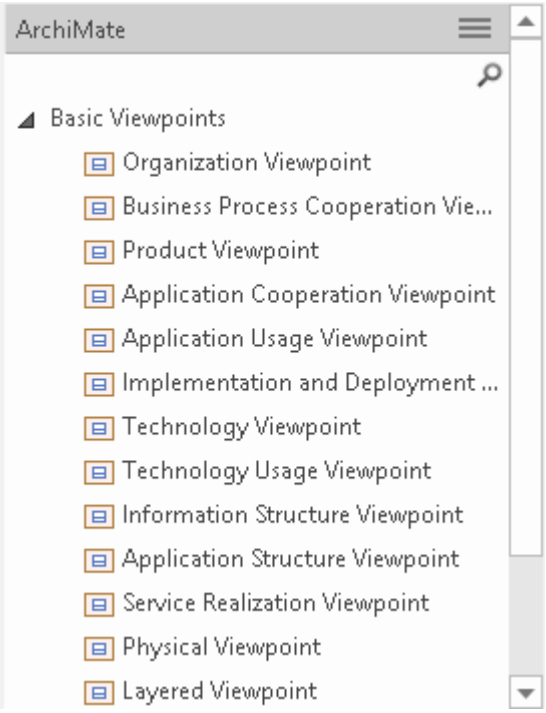


Figure: Showing the basic viewpoints with elements from a number of layers.

Motivation Viewpoints

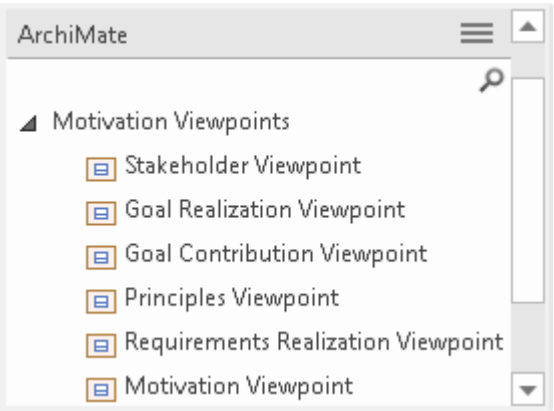


Figure: Showing the viewpoints in the Motivation group that relate elements such as Stakeholders and Drivers.

Strategy Viewpoints

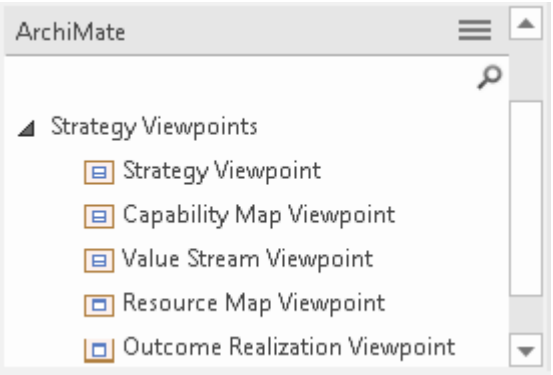


Figure: Showing the viewpoints in the Strategy group containing elements such as Capabilities and Resources.

Implementation and Migration Viewpoints

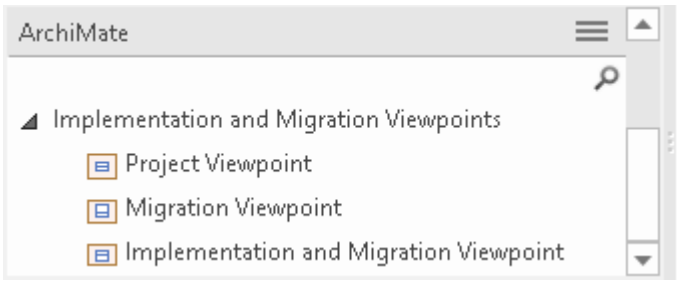
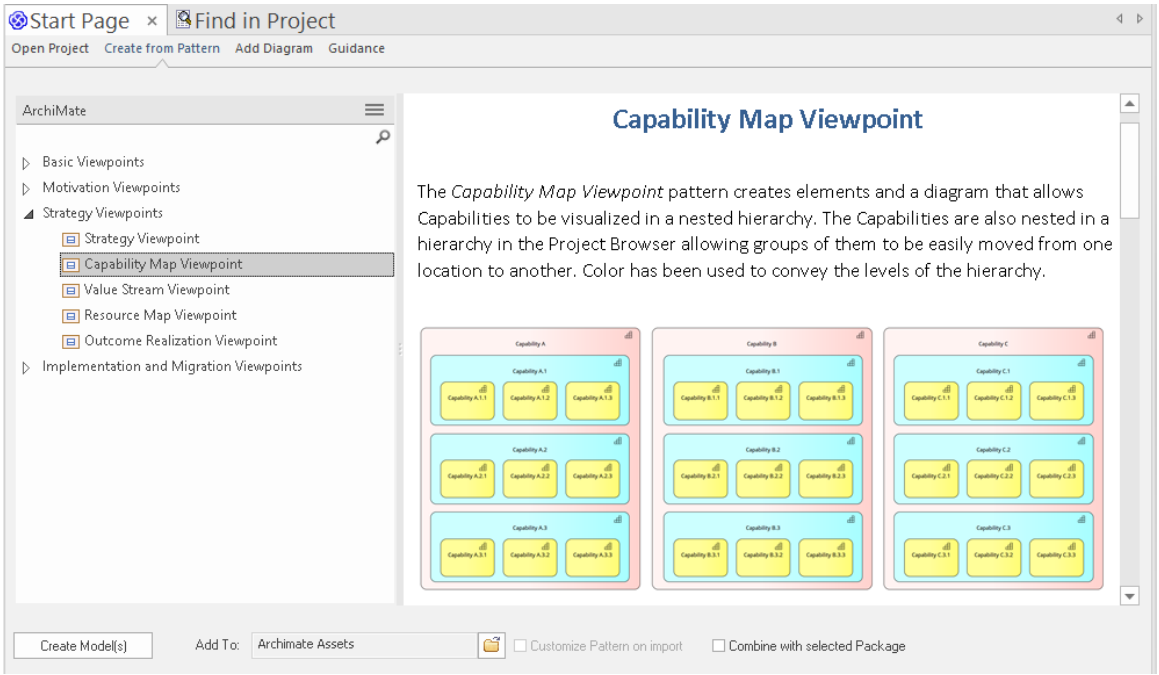


Figure: Showing the viewpoints in the Implementation and Migration group using elements such as Gap and Plateau.

The pattern documentation provides a detailed description of the pattern (in this case a viewpoint) and an example diagram that shows the user what will be generated, including the model content and details of how to use the pattern. This illustration shows the documentation for one of the most important Strategy Viewpoints - the Capability Map viewpoint, which allows Business Architects to create a structured overview of the capabilities of the enterprise.



The documentation clearly describes the viewpoint and provides images that not only show the diagrams that will be created, but also show a screenshot of the Browser window, the elements that would be created and the resulting repository structure. The discussion provides tips that will help the business architect work with the pattern, including next steps and how to manipulate the pattern elements. There is also a list of Help topics and a set of tools that can be used when working with the pattern.

The ArchiMate Language

Enterprise Architect provides a highly compliant implementation of the ArchiMate standard, allowing Enterprise Architects and others to create rigorous and expressive models of enterprise architecture concerns, from the level of stakeholders and their interests right down to the virtual or physical devices that ultimately provide the computational power to run the enterprise and deliver valued services and outcomes to customers.

The language is deliberately pragmatic and has as a design consideration the need for the syntax and symbols to be compact and succinct, but at the same time provide the expressive power to capture and communicate structural and behavioral aspects of the model. The Enterprise Architect platform, with WebEA and Prolaborate, allows teams to create visualizations curated for any stakeholder group, from senior executives down to implementation staff.

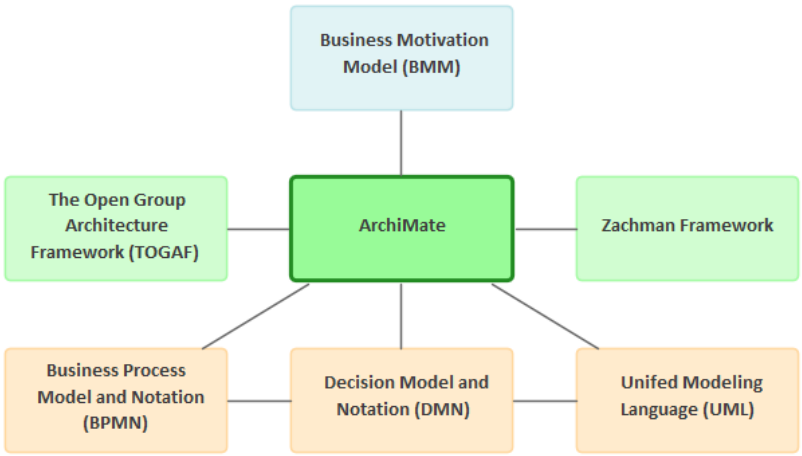


Figure: Showing the relationship between ArchiMate and other modeling languages.

Enterprise Architect implements a large number of other standards, and models produced by other teams can be connected to the Enterprise Architecture models described with ArchiMate, thus creating a tapestry of interwoven threads. For example, ArchiMate deliberately takes a birds-eye view of process models, leaving out much of the details and concentrating on conveying the value and how it relates to Business Functions and Services, Roles and Actors. These models can be connected to more detailed models described the use of Business Process Model and Notation (BPMN). The same is true with respect to application models that, in ArchiMate, are quite high level and intentionally leave out implementation details such as protocols, ports and message flow. These latter details can be modeled using the Unified Modeling Language (UML) and again connected to the ArchiMate Application Components to create expressive models allowing drill-down and drill-up from any modeling context.

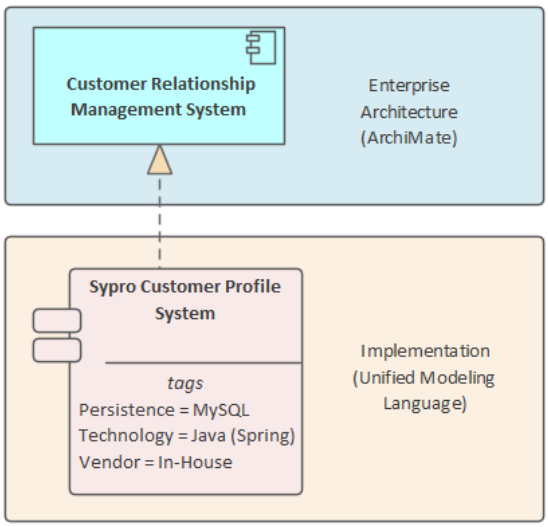


Figure: The relationship created between an ArchiMate application component and a UML component

Language Definitions

Enterprise Architect fully implements the ArchiMate language and supports all the language concepts and definitions, allowing Business, Application, Technology and Security Architects and others - including Geospatial Architects - to create highly expressive and compliant language models, views and other effective visualizations of the enterprises under discussion.

ArchiMate Core Framework

A reference structure used to classify elements of the ArchiMate core language. It consists of three layers and three aspects. Enterprise Architect implements the Core Framework by allowing users to create elements and relationships for each of the three layers including viewpoints. These are all provided as patterns which allow a user to inject a template into their models. Aspects are available in the toolboxes (palettes) of elements and relationships which divide these items into groups. The Aspects are Active Structure, Behavior, Passive Structure and Motivation.

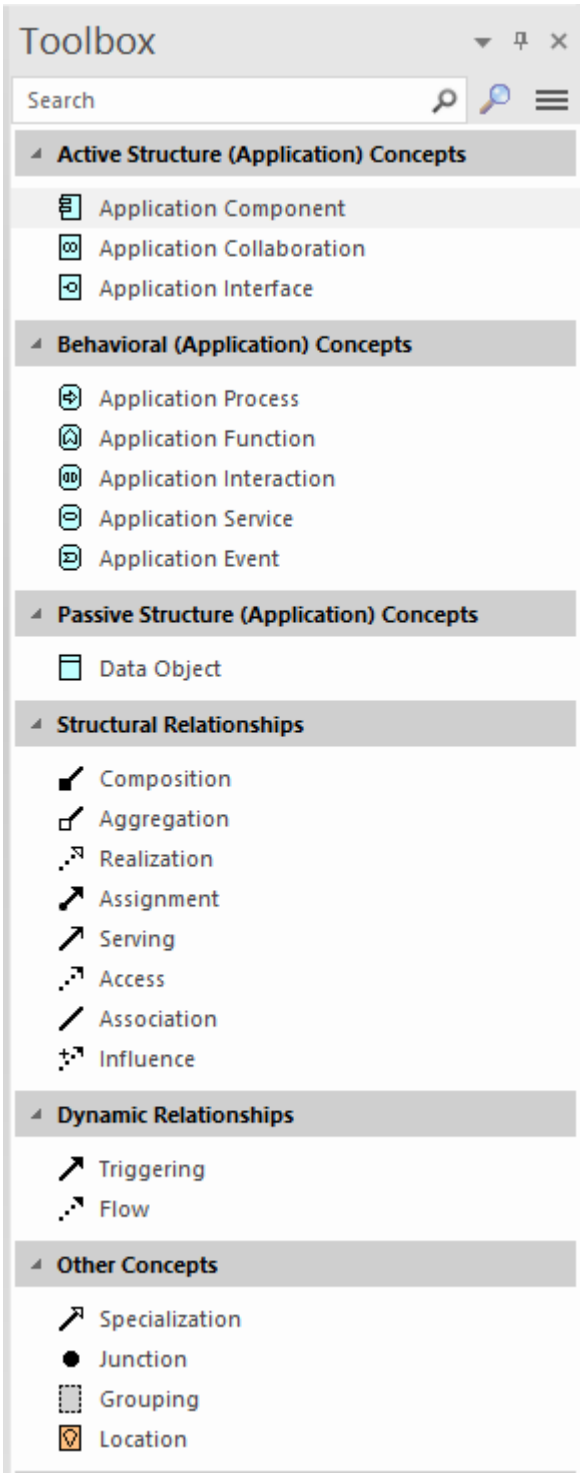


Figure: Showing the elements and relationships grouped by aspect e.g., *Active Structure.*

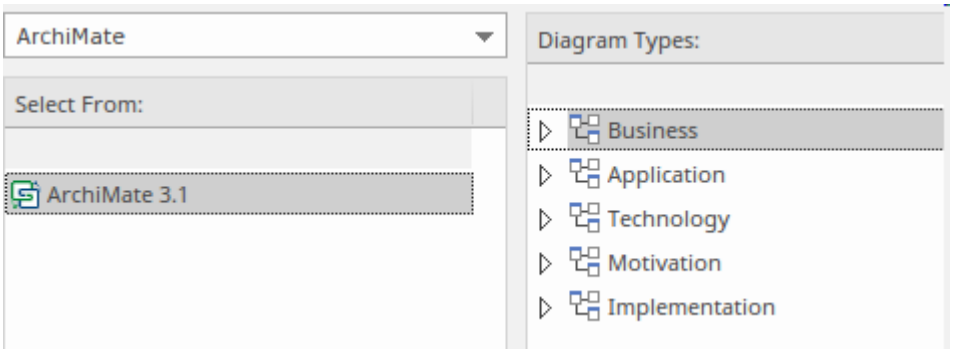


Figure: Showing the ArchiMate diagram dialog where the language layers are visible.

ArchiMate Core Language

The central part of the ArchiMate language that defines the concepts to model Enterprise Architectures has been built into the ArchiMate profile and meta model defined within the MDG Technology inside the tool. The user can access all of the defined ArchiMate language concepts and features, including elements, relationships, relationship connectors and Viewpoints from a series of user-friendly tool features.

Architecture View

A representation of a system from the perspective of a related set of concerns typically relating to a single stakeholder or stakeholder group. Enterprise Architect supports this concept with a range of different product tools and mechanisms. The most important ones are:

- **Diagrams** - any number of diagrams can be created that can display particular elements, properties with selected themes and presentation styles and also restricted using filters
- **Matrices** - elements and their relationships can be displayed in a matrix showing which elements are connected by relationships
- **Textual** - there are a number of tools - including list views, specification views and documents - that allow elements and relationships to be presented in textual form, creating compelling narratives

There is also a Model Wizard pattern tool that allows any of the ArchiMate example Viewpoints to be created. This illustration shows the mechanism to switch between alternative views of the same underlying model content.

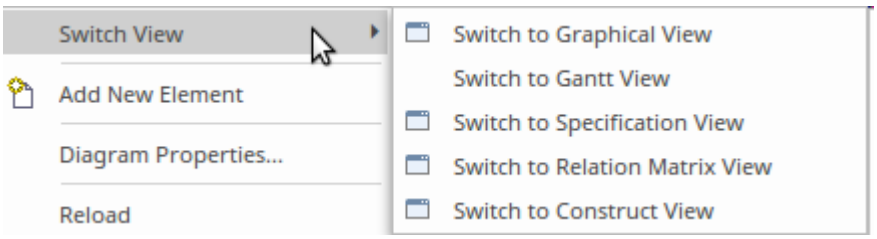


Figure: Showing the tool options to switch views

Architecture Viewpoint

A specification of the conventions for a particular kind of architecture view. Enterprise Architect provides a productivity and compliance tool called Model Wizard patterns that allows all the example ArchiMate viewpoints to be created from well defined and described patterns built into the ArchiMate perspective. This diagram shows the result of using the Wizard to create a Technology Usage Viewpoint.

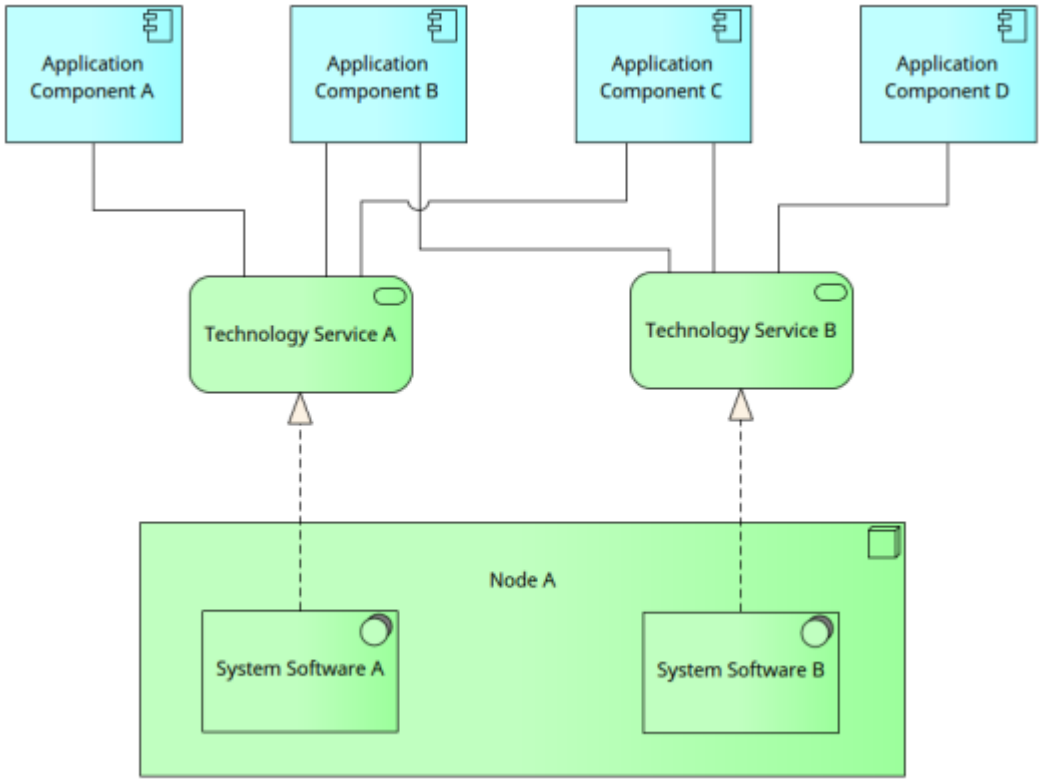


Figure: Showing the Technology Usage Viewpoint

Aspect

The tool supports the concept of Aspects, which fundamentally divide the corpus of elements into syntactic groups based on grammatical roles. The Aspects allow the modeler and the viewer to understand the model and diagrammatic constructs in terms of the role they play. The Aspects are Active Structure, Behavior, Passive Structure and Motivation. These are visible throughout the tool's implementation of ArchiMate in the Diagram Toolbox, which make new elements available to be added to the model and diagrams.

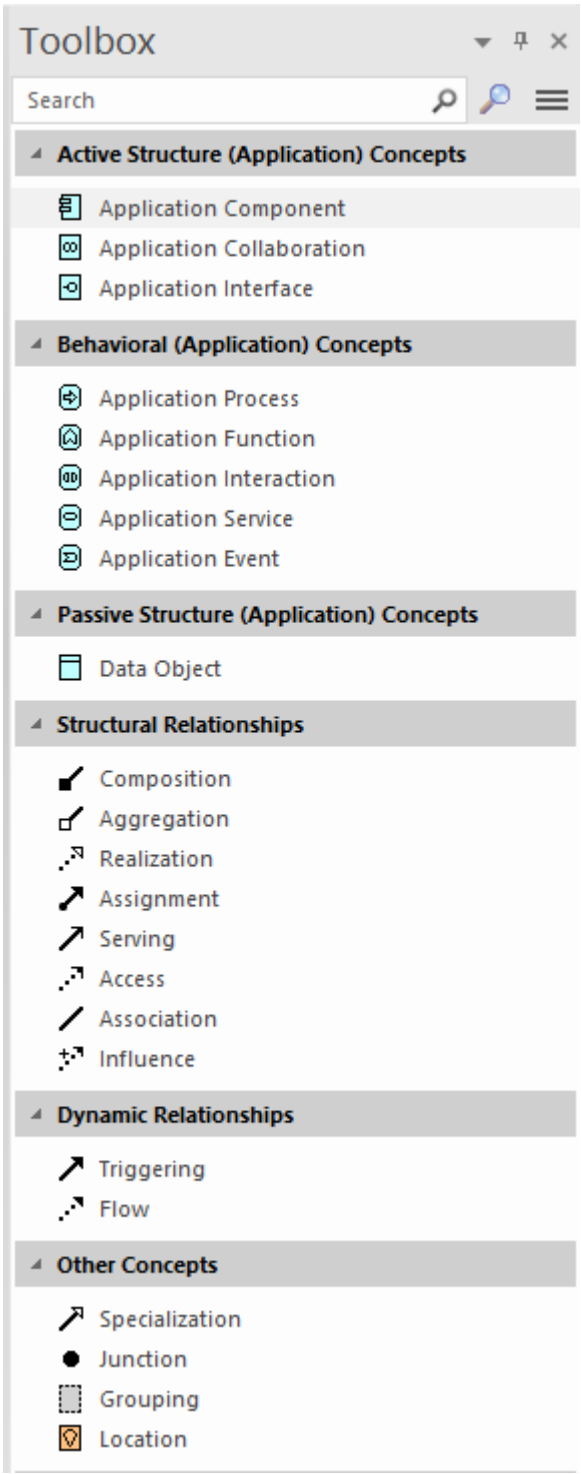


Figure: Showing the elements and relationships grouped by aspect e.g., *Active Structure, Behavior and Passive Structure elements.*

Attribute

Enterprise Architect supports a number of ways of adding properties to elements and relationships. Elements can have two different types of property:

- Built-in Properties: *e.g. Name, Notes, Alias, Author*

- Custom Properties: *e.g. Business Owner, Technical Owner, End-of-Life Date*

Project	
Author	Pauline Dean
Package	Examples
Phase	1.0
Complexity	Easy
Created	2/03/2020 9:37:25 AM
Modified	10/03/2020 1:16:52 PM
Language	<none>
Filename	C:\users\PDean\Documents\applications inventory
GUID	{AE38B5C8-5430-4a35-ACC4-9694CB148F2F}
WebEA	https://some-cloud.com.au?m=4&o=AE38B5C8-5430-

Figure: Showing the element Properties window

It is common for an Enterprise Architecture function or practice to add organization-specific properties that help in the analysis of business, application and technology architectures. These can be added as Tagged Values either directly to an element, or more robustly using the profile system that allows the creation and augmentation of elements as a new type that results in a fully compliant ArchiMate element with the additional properties.

Composite Element

The tool supports the creation of the two Composite Elements namely: Grouping and Location which allow aggregation relationships to other elements typically from multiple aspects or layers of the language.

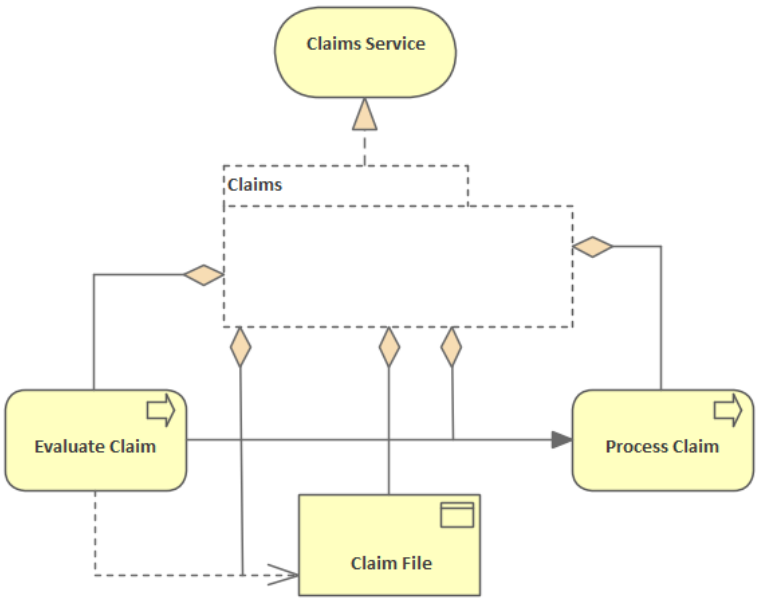


Figure: Showing the Grouping element

Core Element

Enterprise Architect supports all the core elements defined in the ArchiMate specification, in all aspects and all layers including the extensions.

Element

The tool supports the creation and modification of all the ArchiMate elements across all layers and aspects, and provides mechanisms to place these elements onto diagrams that comply with the example viewpoints, including allowing them to participate in relationships with other elements (and, in some cases, other relationships) in accordance with the defined combinational rules. Elements, along with Relationships and Relationship Connectors, are the most primitive type of concept in the ArchiMate metamodel and are used to define and describe the constituent parts of Enterprise Architectures and their unique set of characteristics.

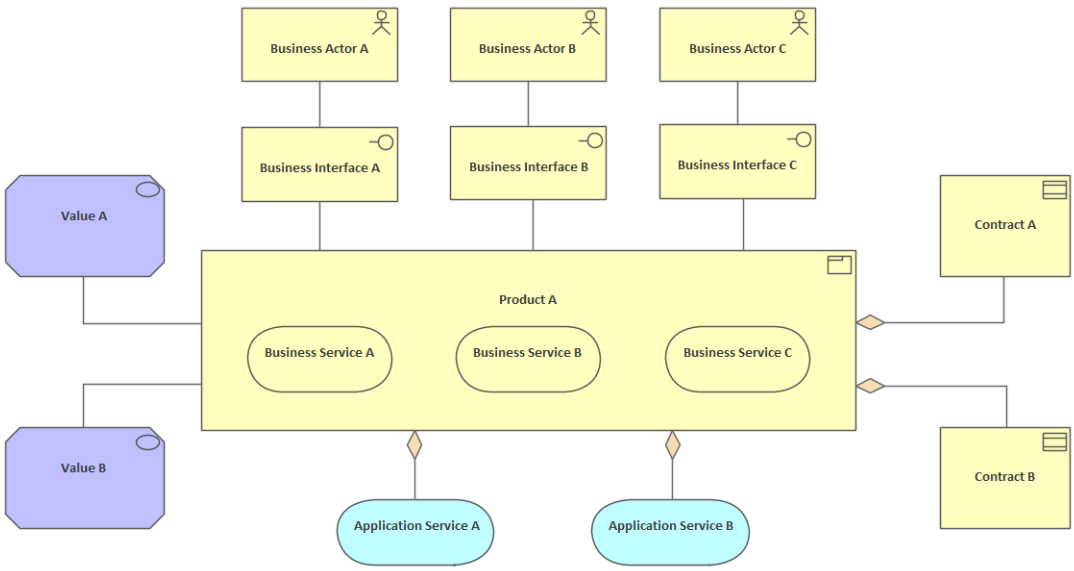


Figure: Showing the *Product* Viewpoint

In this diagram we see elements from a number of different layers, using a number of different relationships, including nesting as an alternative. Some of the elements are presented using their rectangular style and others (e.g. Application Services) are presented using their icon style. The style can be toggled in Enterprise Architect to suit the audience and stakeholders.

Layer

Enterprise Architect is an effective platform for architectural description and visualization, and it supports a variety of domain architects that are represented in the language and the tool by the concept of layers. These domain-specific models can be articulated with each other without ambiguity and are visible in the tool in the form of diagrams and their accompanying Toolbox pages.

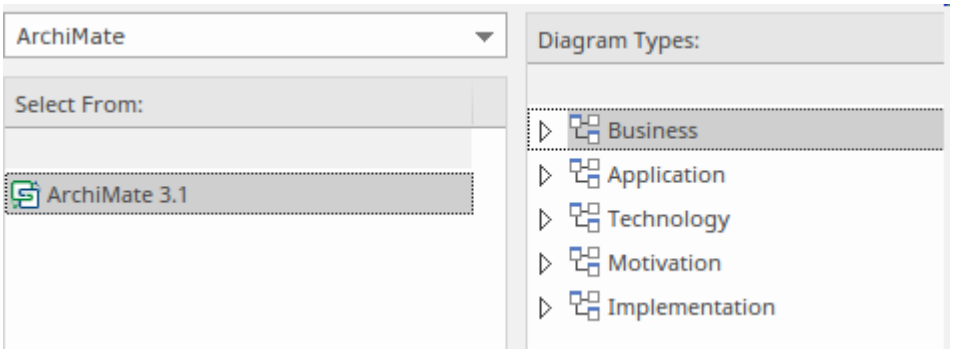


Figure: Showing the ArchiMate diagram dialog where the language layers are visible.

Model

The tool provides a repository based solution which includes the ability to create any number of models either in a single repository for the purposes of collaboration and model sharing or in separate and isolated repositories for the purpose of isolation or the separation of concerns. Regardless of how the models are organized architects and other stakeholders can access the models via the Internet in desktop and browser-based tools.

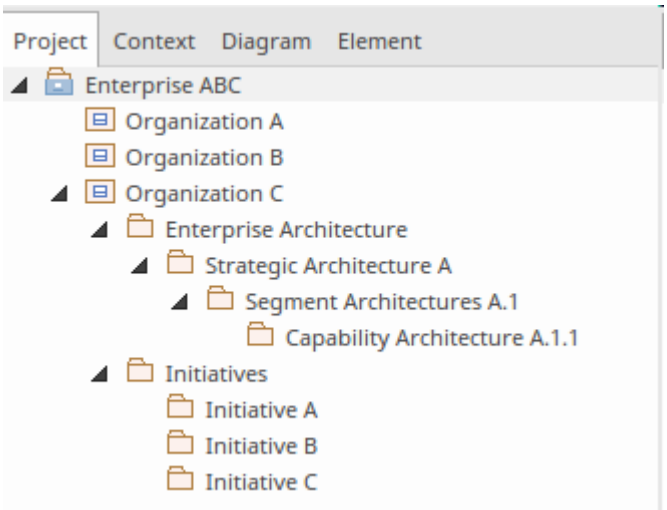


Figure: Showing the Package structure in the Browser Window.

Relationship

Enterprise Architect supports the creation and modification of all the ArchiMate relationships and relationship connectors across all layers and aspects, and provides mechanisms to place these relationships onto diagrams, thus connecting elements (and in some cases other relationships) in compliance with the specification. The tool has a productivity tool called the Quick Linker, which allows relationships to be created by dragging from the corner of a source element to a target element. The Quick Linker restricts the available relationships for a source-target pair to the set of permissible relationships based on the specification. These are defined in the Sparx Systems ArchiMate metamodel that underpins the implementation and ArchiMate Technology.

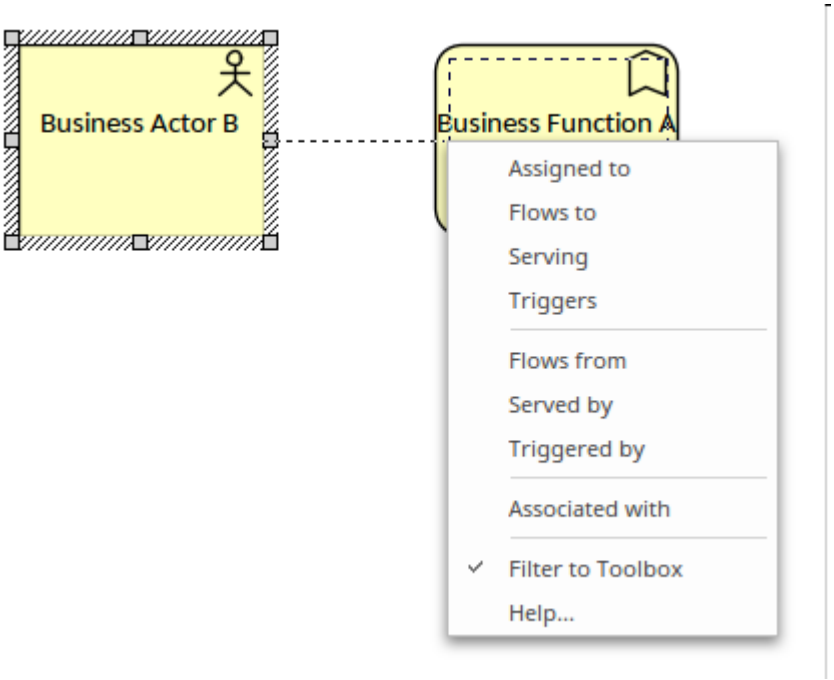


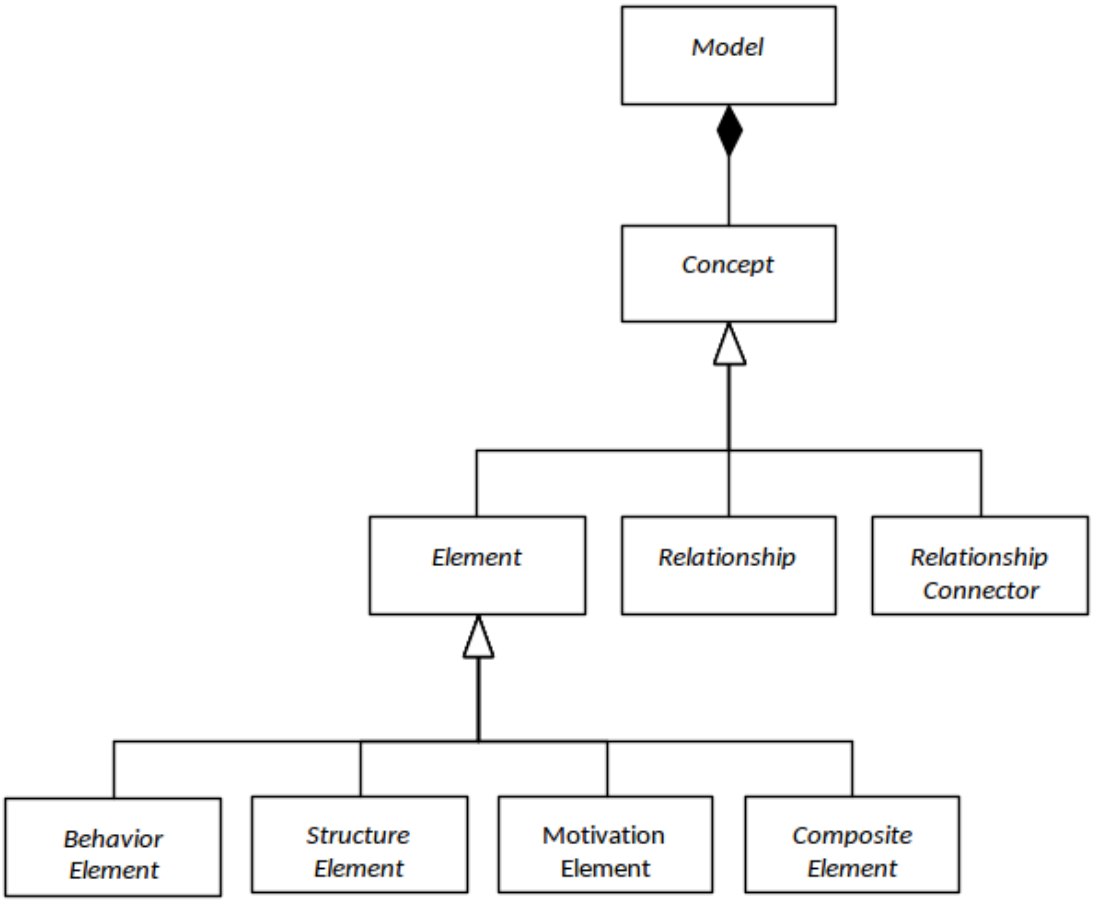
Figure: Example Quick Linker menu showing the restricted set of relationships for the elements

Language Structure

When you create architectures in Enterprise Architect you are fundamentally creating ArchiMate Models that are made up of a collection of these concepts:

- Elements
- Relationships
- Relationship Connectors

These are indicated in this Metamodel diagram.



Elements of one of four types can be created and will appear in the Browser window:

- Structure Elements
- Behavior Elements
- Motivation Elements
- Composite Elements

Generic Metamodel

The main hierarchy of behavior and structure elements of the ArchiMate language is shown in this diagram. All the elements in this model have italic names, indicating that they are abstract and do not themselves participate in models. They are represented as independent of the layers of the framework as they can apply across all layers. The most fundamental division of the language elements is into Behavior and Structure elements.

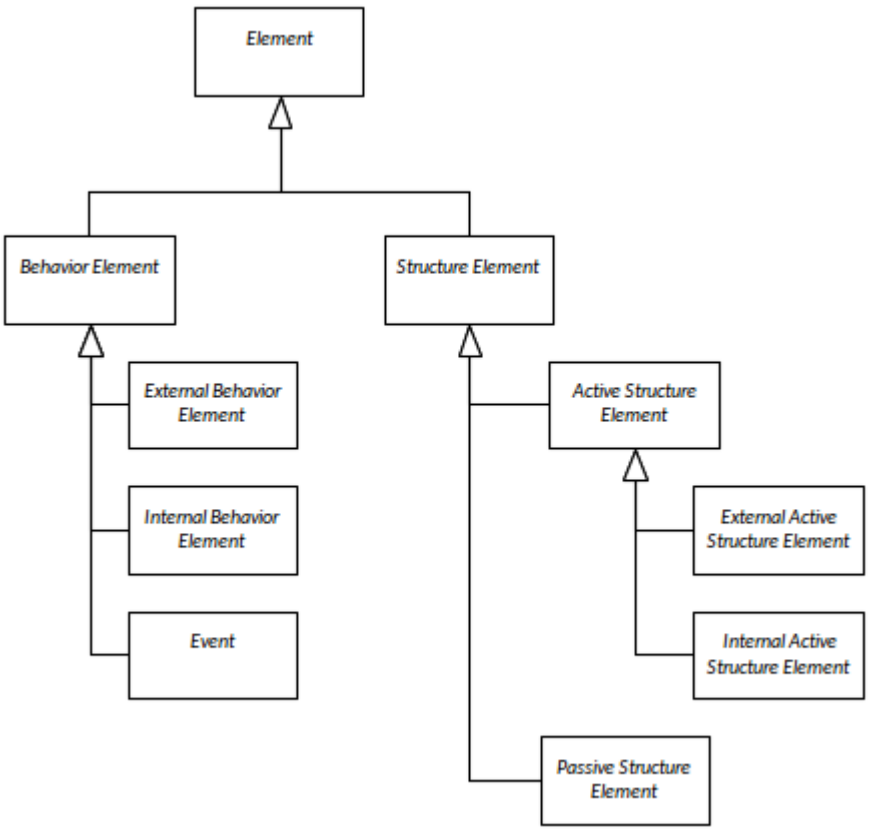


Figure: Showing the fundamental hierarchy of the ArchiMate elements.

Elements such as an Application Component - which is a type of Active Structure element - and Application Process - which is a type of behavior element - are added to models of architecture.

Behavior and Structure Elements

The main hierarchy of behavior and structure elements of the ArchiMate language is implemented in Enterprise Architect. The behavior elements are analogous to verbs and the structure elements to nouns in a natural language.

Active Structure - elements are visible in the Enterprise Architect Toolbox pages and represent a subject that can perform behavior.

Behavior - elements are visible in the Enterprise Architect Toolbox pages and represent the dynamic aspects of an enterprise.

Passive Structure - elements are visible in the Enterprise Architect Toolbox pages and represent items that are worked on by active structure elements. Data, and Information objects are examples of passive structure elements.

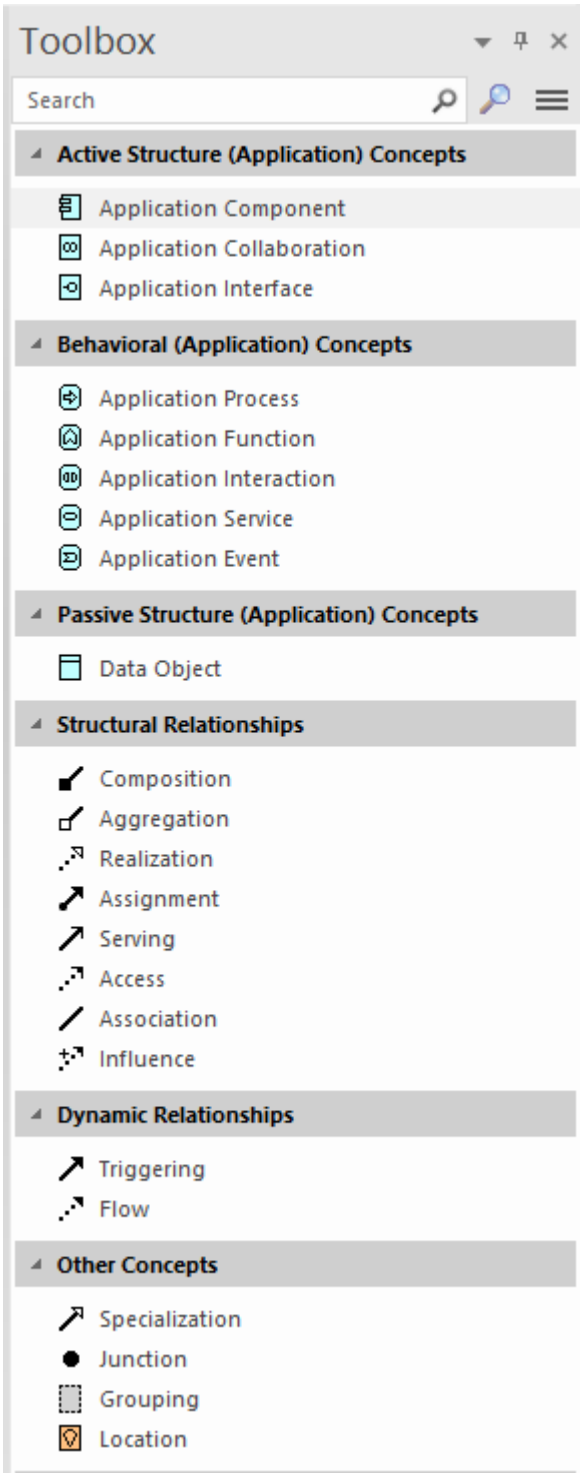


Figure: Showing the Application Layer elements and relationships grouped by aspect e.g., *Active Structure, Behavior and Passive Structure elements.*

Specializations of Structure and Behavior Elements

An architect can use composition and aggregation between processes, functions and interactions.

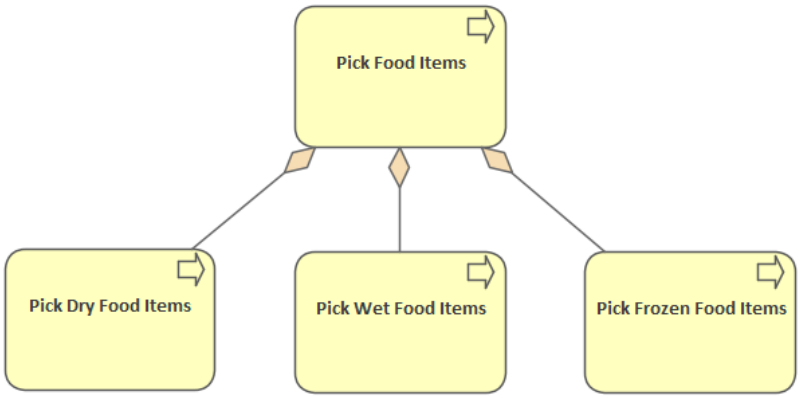


Figure: Showing the aggregate relate between a number of business processes modeling the picking of items for customer orders.

The collective nature of a behavior can be made either *implicit* (several active structure elements assigned to the same internal behavior) or *explicit* through the use of a collective internal behavior (interaction) that is performed by multiple collaborating active structure elements.

Summary of Structure and Behavior Elements

Enterprise Architect implements the core elements from the meta model and provides layer specific representations of these elements. For example a there is a business external active structure element represented by a Business Interface and a Technology *external active structure* element represented by a Business Interface. There is also a technology *external active structure* element represented by a technology Interface.

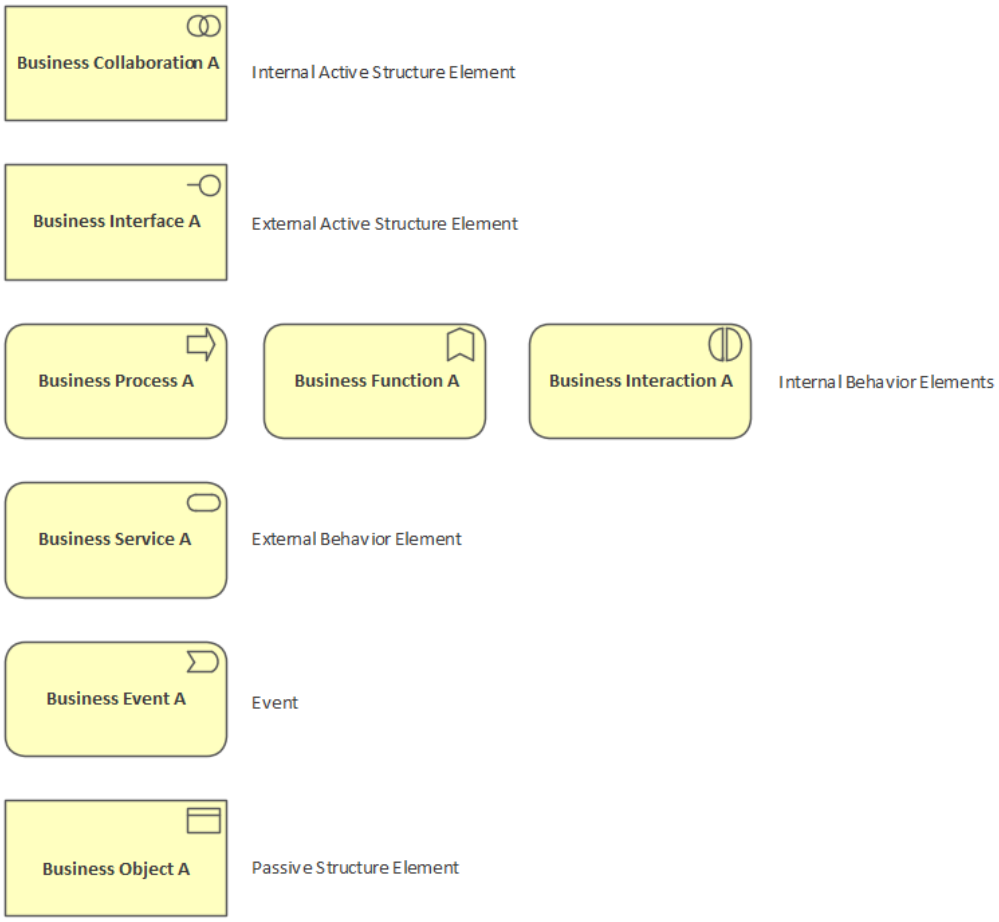


Figure: Showing a range of concrete Business Elements that represent the abstract core elements.

Motivation Elements

The motivation aspect allows users to create models of the forces that drive and design the operation of the enterprise. Enterprise Architect provides a palette of these motivation elements, including stakeholder, value, meaning, driver, assessment, goal, outcome, principle, and requirement, which also includes its subtype constraint.

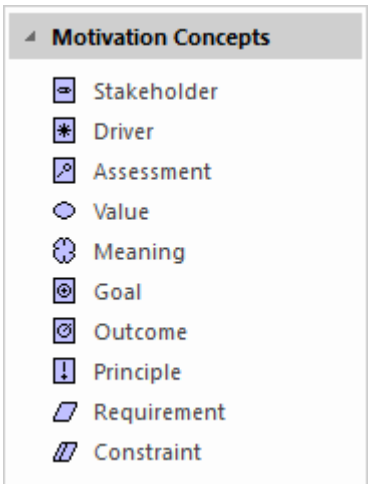


Figure: Showing the *Motivation Toolbox Elements*

Composite Elements

Enterprise Architect implements the Archimate Composite elements and allows these elements to be aggregated or compose other composite elements. For example, an architect could model a large warehouse facility as a location and create composite locations that represent the position of specific stockpiles within the containing location.

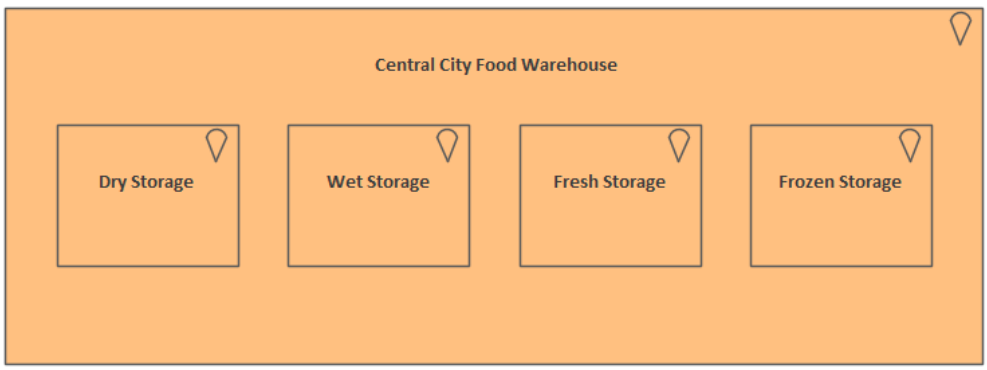


Figure: Showing *Composite Locations in a Food Warehouse model with multiple storage locations for different types of foods.*

Elements

Enterprise Architect provides a rich and effective user interface for working with elements. It allows all the ArchiMate elements across all aspects and layers to be created, managed and visualized. This includes these types of element corresponding to the ArchiMate Toolbox pages.

1. Motivation (Aspect) Elements
2. Strategy (Layer) Elements
3. Business (Layer) Elements
4. Application (Layer) Elements
5. Technology (Layer) Elements
6. Implementation and Migration (Layer) Elements
7. Composite Elements

Changing Element Presentation

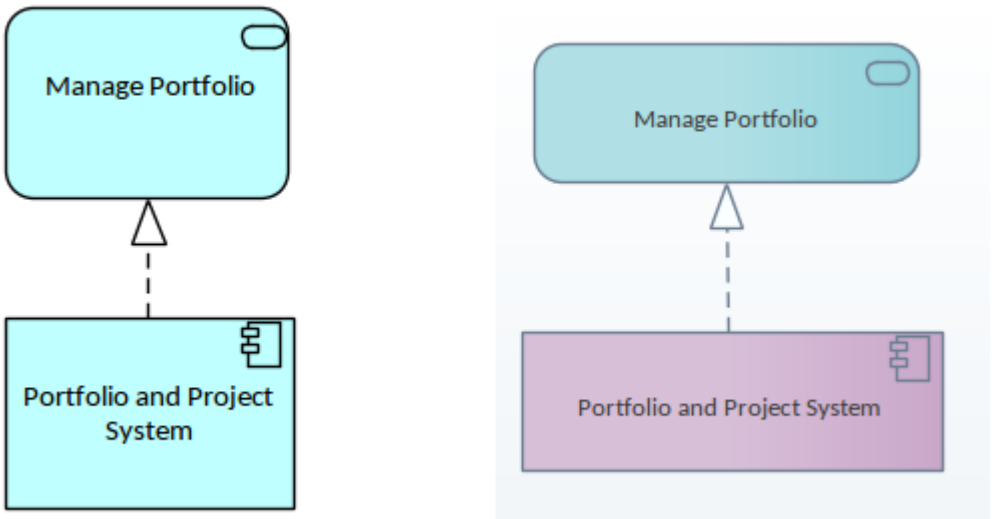
An Enterprise Architect will typically need to provide different visualizations for different groups of stakeholders, and a diagram that is suitable for one group might not be appropriate for another. Enterprise Architect provides a wide range of ways for working with elements and diagrams, including how the elements are presented and visualized in different diagrams and different windows within the product.

Enterprise Architect has great flexibility and, because the tool is implemented using a robust relational database back-end, when changes are made to a single element on a diagram or any other view the changes can be automatically applied to all places in diagrams or other windows where the element appears. This applies to element properties, name changes and other fundamental aspects of the element. When the appearance (such as color and style) of an element is changed a modeler has a choice of:

- Changing the diagram Object style (just applies to the current diagram)
- Changing the default element appearance (applies to the current diagram AND on all other diagrams)

Changing Size, Proportion and Color

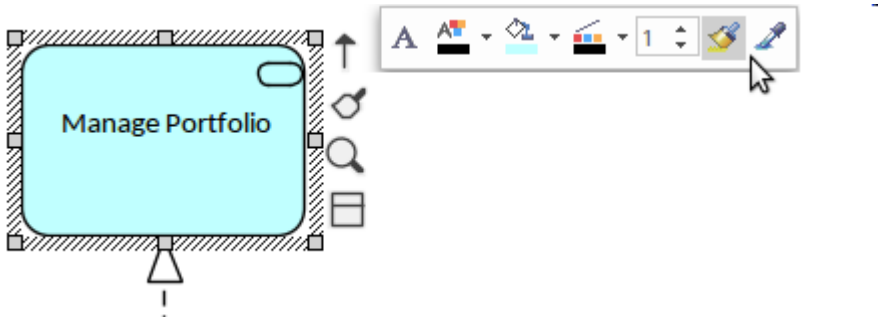
Any of the standard ArchiMate elements can be changed to different sizes, proportions and colors without affecting the meaning or syntax of the diagrams. Enterprise Architect provides great flexibility for changing both the elements and the diagram styles and themes. This illustration shows two different diagrams where the same elements have been rendered in different sizes, proportions and colors.



Changes on a Single Diagram

Changes can be made to the style of an object (element) on a single diagram; once the diagram is opened and the object is selected there are a number of places where the element style can be changed. These changes will only apply to the selected element in the diagram and other diagrams that contain the element will not be changed in any way. In the next section we will see how, if required, the changes can be made to update all other diagrams.

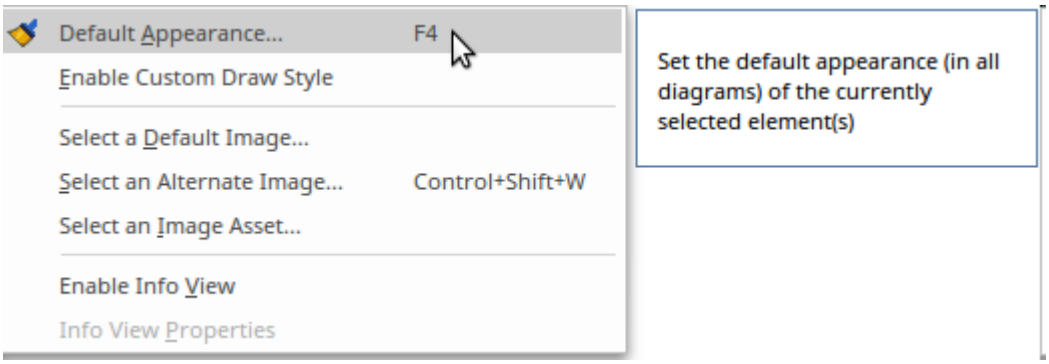
The options to change the style are available from a number of locations, including the ribbons and the element quick style icon that is displayed when the element is selected in the diagram. This illustration shows the options available from the diagram.



Changes on All Diagrams

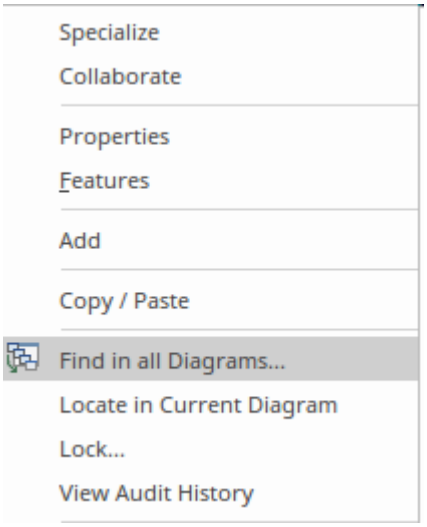
Changes can be made to the style of an object (element) on all diagrams; once the diagram is opened and the object is selected there are a number of places where the element style can be changed. These changes will apply to the selected element in the diagram, and all other diagrams that contain the element will also be changed in any way.

The options to change the style are available from a number of locations, including the ribbons and the element quick-style icon that is displayed when the element is selected in the diagram. This illustration shows the options available from the diagram.

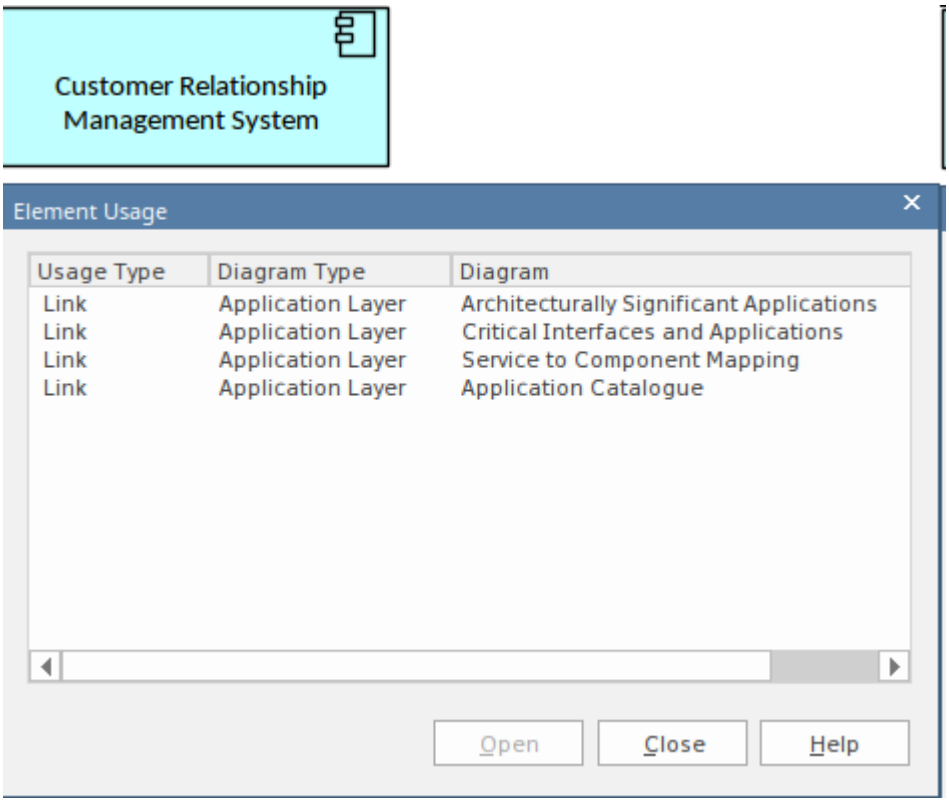


Element Usage

Elements from the repository can participate in any number of views. They also have significance in the null case where they do not appear in any views. The inclusion of a variety of elements on a given diagram and the inclusion of a given element on multiple diagrams are what gives the language its expressive power. Enterprise Architect supports all these language mechanisms and also allows you to locate the different diagrams that contain a particular element.



Selecting this option provides a list of all the diagrams that contain the element, allowing you to hyperlink and open any of the listed diagrams. This provides a convenient way to understand the basis for the element's usage, as the diagram's type and name are included in the table of usages.

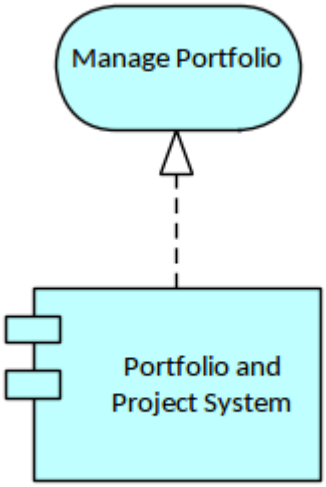


Rectangular and Icon Views

The ArchiMate language provides the flexibility of displaying a number of elements either as rectangles with a small icon in the top right hand corner, or in Icon view where the whole element takes on the Icon shape. As an Icon, wherever possible the name of the element is placed inside the shape, but with some Icons there is no room and the name is positioned underneath the element. Not all elements have an Icon representation and in these cases the icon view cannot be selected.

The tool makes it easy to toggle between the two views, but care must be taken as the geometry of the element will

change as the icons are typically smaller than the rectangular representation.



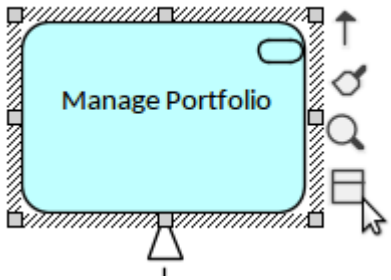
To show an element using rectangle notation, either:

- Click on the element in the diagram and on the rectangle icon that displays on the right of the element, or
- Right-click on the element on the diagram and select the 'Appearance | Use Rectangle Notation' context menu option

This displays the element as a rectangle, by default displaying an ArchiMate icon in the top right corner of the element (where the language defines an icon).

The setting only applies to the selected element, and can be toggled on and off either by:

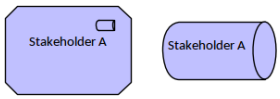
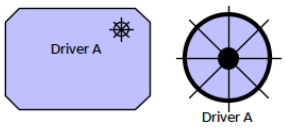
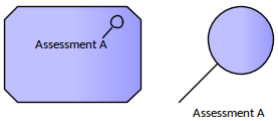
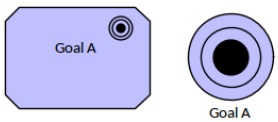
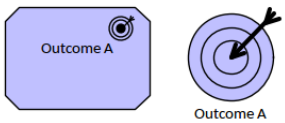
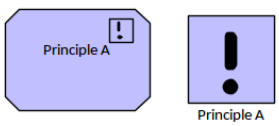
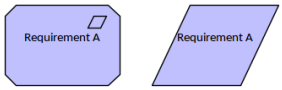
- Clicking on the icon again
- Deselecting the context menu option, or
- Selecting the reciprocal context menu option such as 'Use Circle Notation' or 'Use Actor Notation'

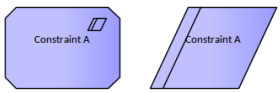
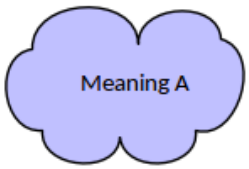
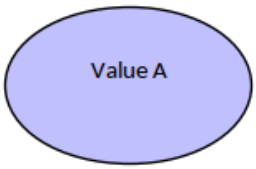


Motivation Elements

Motivation elements are used to model the motivations, or reasons, that guide the design or change of an Enterprise Architecture.

Table of Motivation Elements

Stakeholder 	A *stakeholder* represents the role of an individual, team, or organization (or classes thereof) that represents their interests in the effects of the architecture. The image demonstrates both standard and alternative element notation.
Driver 	A *driver* represents an external or internal condition that motivates an organization to define its goals and implement the changes necessary to achieve them.
Assessment 	An *assessment* represents the result of an analysis of the state of affairs of the enterprise with respect to some driver.
Goal 	A *goal* represents a high-level statement of intent, direction, or desired end state for an organization and its stakeholders.
Outcome 	An *outcome* represents an end result.
Principle 	A *principle* represents a statement of intent defining a general property that applies to any system in a certain context in the architecture.
Requirement 	A *requirement* represents a statement of need defining a property that applies to a specific system as described by the architecture.

Constraint 	A *constraint* represents a factor that limits the realization of goals.
Meaning 	*Meaning* represents the knowledge or expertise present in, or the interpretation given to, a concept in a particular context.
Value 	*Value* represents the relative worth, utility, or importance of a concept.

Motivation Example Diagrams

Using Enterprise Architect a modeler can create any number of diagrams using the ArchiMate diagrams and Toolbox palettes. The motivation elements can be connected together to create expressive narratives that describe an Enterprise Architecture and the motivation for making business or technology changes articulated in other parts of the architecture. These diagrams have been taken from the Open Group's ArchiMate Specification.

Goal, Outcome, Principle, Requirement, and Constraint

This diagram is concerned with the relationship between Goals, Principles, Requirements, and Constraints. The goal “Improve Profitability of Service Offering” is realized by the outcome “Increased Profit by 10% in Next Fiscal Year” and “Reduced Cost of Customer Acquisition by 25%”. We see the use of the influence relationship between outcomes with the indicator of (+) or (-) effect. The outcomes are in turn influenced by a number of principles and the principles are realized by a number of requirements. We also see the use of the OR Junction in the bottom left of the diagram.

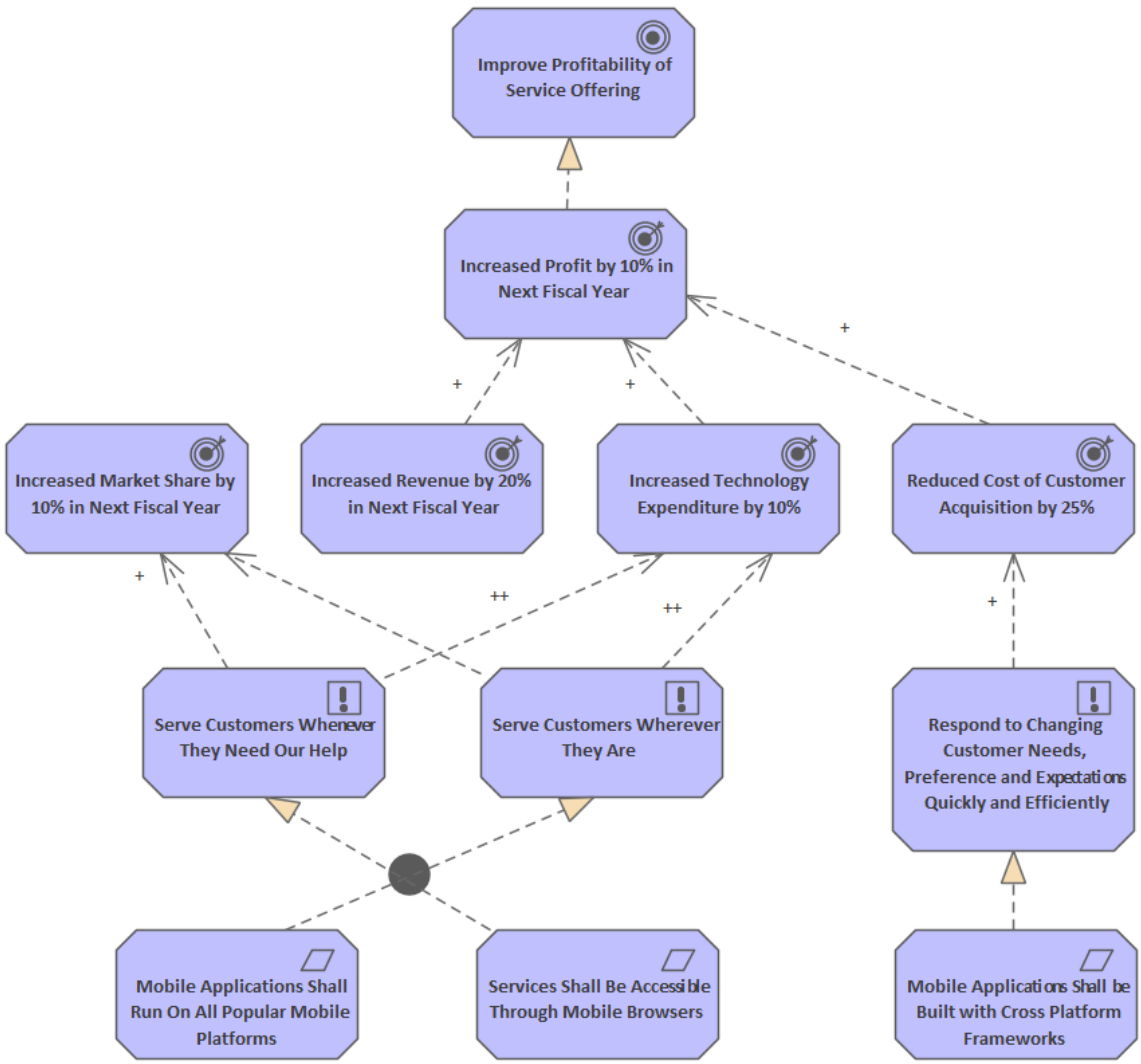


Figure: Example diagram from the ArchiMate Specification showing how Goals, Outcomes, Principles and Requirements (including constraints) are modeled.

Stakeholder, Driver, and Assessment

This diagram is concerned with Drivers and the Stakeholders that are concerned about them, which in this example are at the executive level. We see that the Driver entitle Profitability has been decomposed into two other drivers namely: Revenue and Costs. Assessments have been associated with the Drivers which paint a clear picture of the state of the organization. Influence relationship (which can be drawn between any two motivation elements) show how the assessments are related to each other. The (+) indicating a positive influence and (-) indicating a negative influence. This is example 18 in the Motivation Chapter of the specification.

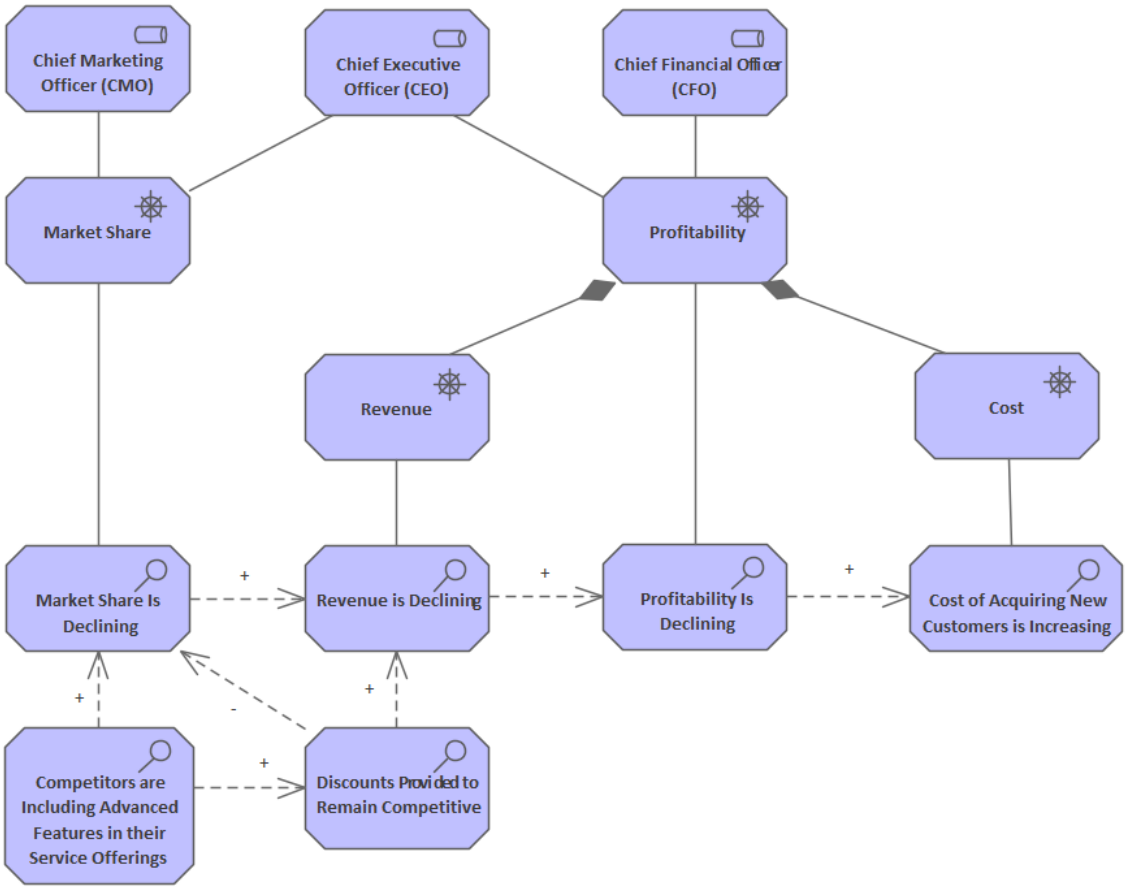


Figure: Example diagram from the ArchiMate Specification showing how Stakeholders, Drivers and Assessments are modeled.

Meaning and Value

This diagram is concerned with stakeholders and the value they derive from the system. The notification message has been specialized into three different types. Meanings have been associated with the messages:

- A “Confirmation Of Receipt" message” has the meaning “Claim Has Been Received”
- A “Review Complete" message has the meaning “Claim Review Complete”
- A “Payment Complete" message” has the meaning “Claim Has Been Paid”

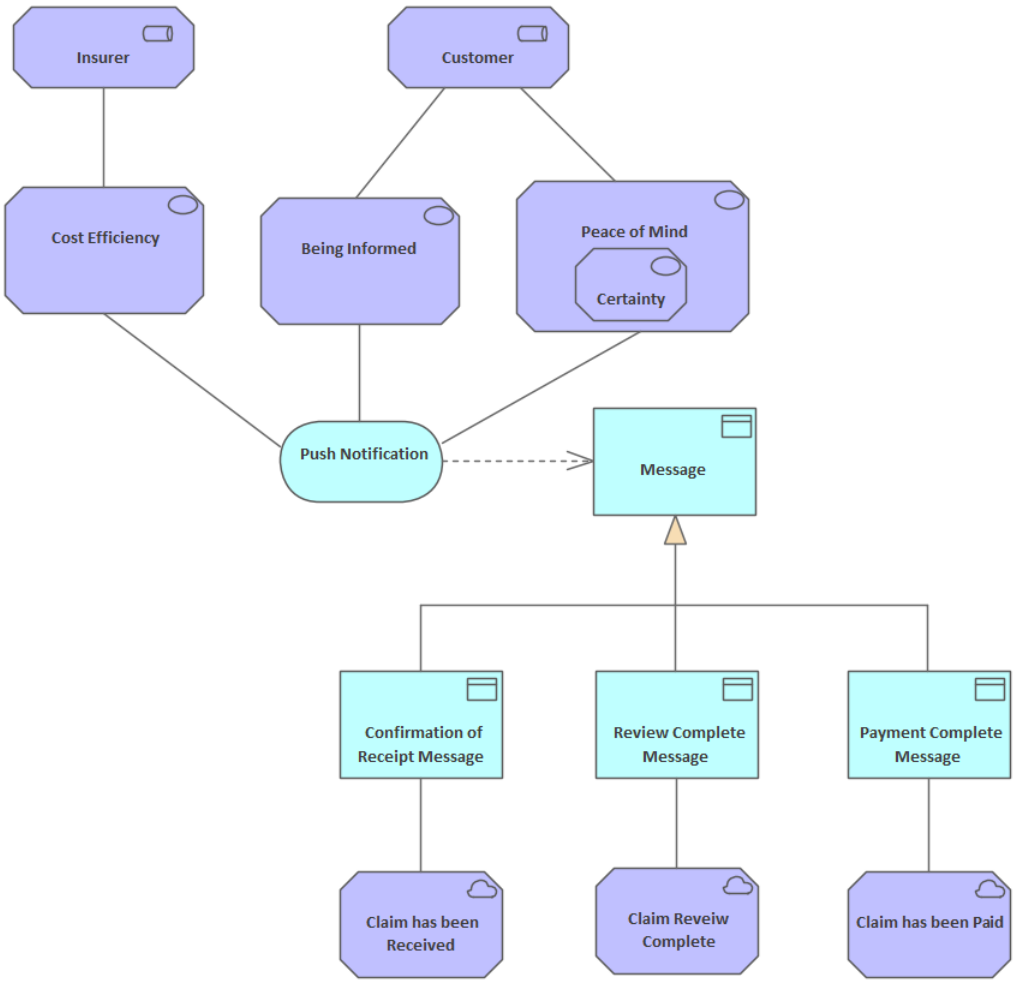
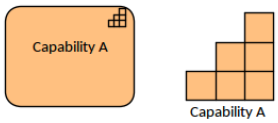
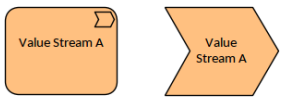
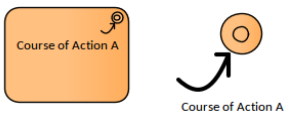
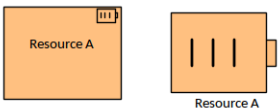


Figure: Example diagram from the ArchiMate Specification showing how Meaning and Value are used.

Strategy Layer Elements

Strategy elements are used to model the strategic consideration of an organization or its parts.

Table of Strategy Elements

Capability 	A *capability* represents an ability that an active structure element, such as an organization, person, or system, possesses.
Value Stream 	A *value stream* represents a sequence of activities that create an overall result for a customer, stakeholder, or end user.
Course of Action 	A *course of action* represents an approach or plan for configuring some capabilities and resources of the enterprise, undertaken to achieve a goal.
Resource 	A *resource* represents an asset owned or controlled by an individual or organization.

Strategy Layer Example Diagrams

Capability, Resource,and Course of Action

This diagram is concerned with an overarching Goal to Increase Profit that is made up of two other Goals (these could be considered to be Objectives) namely Decrease Costs and Increase Revenue. Then two Outcomes are modeled that influence the delivery of the Goals. The business architect has modeled two Course of Action elements that influence these Outcomes. The Course of Action elements are Realized by two Capabilities one of which has two Resources Assigned at Headquarters.

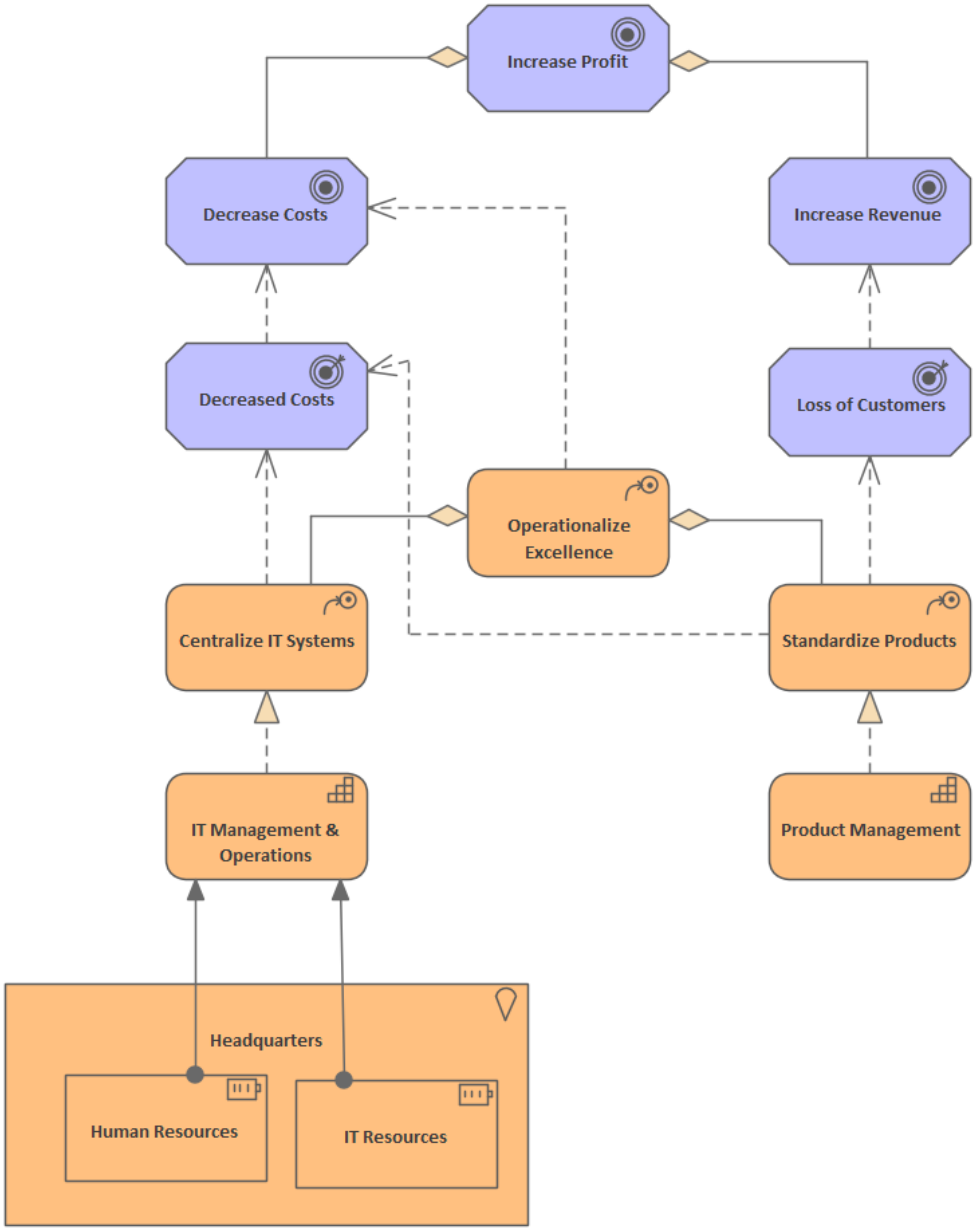
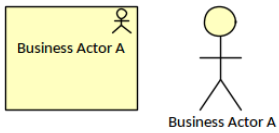
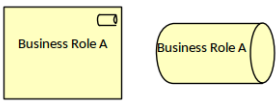
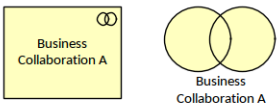
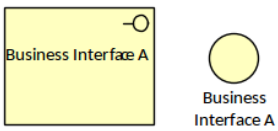
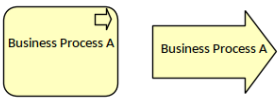
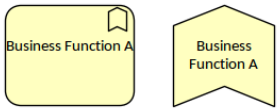


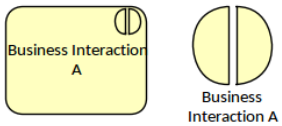
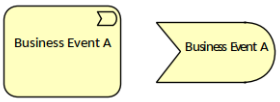
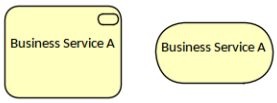
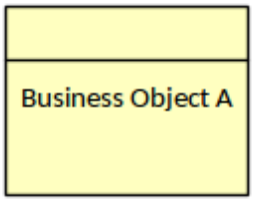
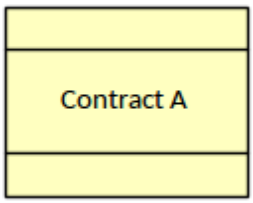
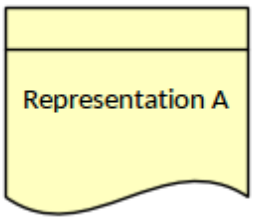
Figure: Example diagram from the ArchiMate Specification showing Capabilities, Courses of Actions and Resources.

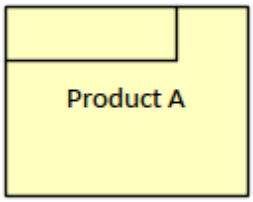
Business Layer Elements

Business elements are typically used in conjunction with strategy elements to model the business architecture of an organization or its parts.

Table of Business Layer Elements

Business Actor 	A *business actor* represents a business entity that is capable of performing behavior.
Business Role 	A *business role* represents the responsibility for performing specific behavior, to which an actor can be assigned, or the part an actor plays in a particular action or event.
Business Collaboration 	A *business collaboration* represents an aggregate of two or more business internal active structure elements that work together to perform collective behavior.
Business Interface 	A *business interface* represents a point of access where a business service is made available to the environment.
Business Process 	A *business process* represents a sequence of business behaviors that achieves a specific result such as a defined set of products or business services.
Business Function 	A *business function* represents a collection of business behaviors based on a chosen set of criteria (typically required business resources and/or competencies), closely aligned to an organization, but not necessarily explicitly governed by the organization.
Business Interaction	A *business interaction* represents a unit of collective business behavior performed by (a collaboration of) two or more business actors, business roles, or business collaborations.

	
Business Event 	A *business event* represents an organizational state change.
Business Service 	A *business service* represents explicitly defined behavior that a business role, business actor, or business collaboration exposes to its environment.
Business Object 	A *business object* represents a concept used within a particular business domain.
Contract 	A *contract* represents a formal or informal specification of an agreement between a provider and a consumer that specifies the rights and obligations associated with a product and establishes functional and non-functional parameters for interaction.
Representation 	A *representation* represents a perceptible form of the information carried by a business object.
Product	A *product* represents a coherent collection of services and/or passive structure elements, accompanied by a contract/set of agreements, which is offered as a whole to (internal or external) customers.

 Product A	

Business Layer Example Diagrams

Business Objects, Representations and Contract

This diagram is concerned with the Business Passive Structure elements. It shows how an Insurance Claim can be Realized by a number of different representations. An Insurance Policy modeled using a Contract element is Realized by a Policy Summary, which is a Representation. All of the representations include this element, as indicated by the use of the Aggregation relationship.

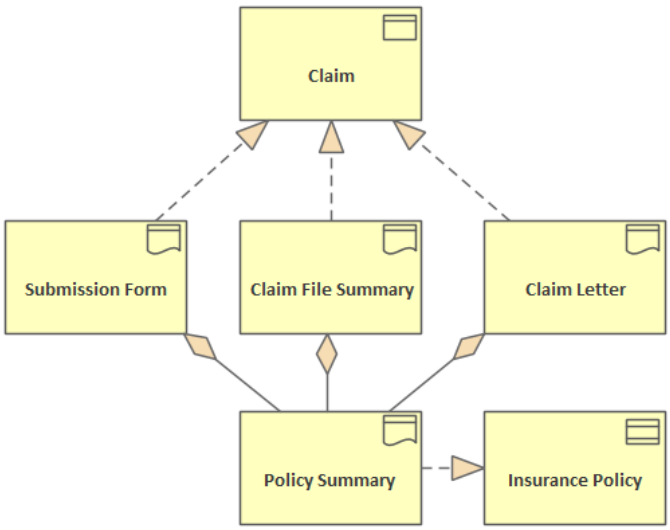
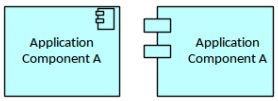
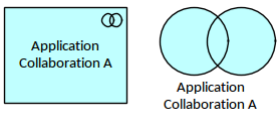
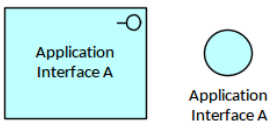
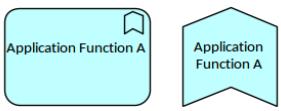
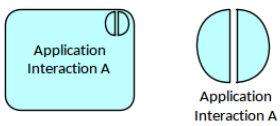
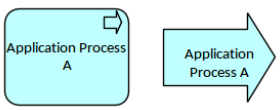
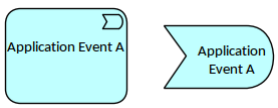


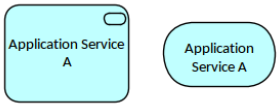
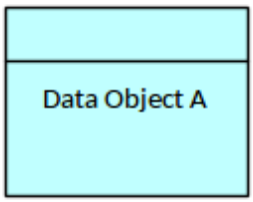
Figure: Example diagram from the ArchiMate Specification, showing Business Objects, Representations and Contracts

Application Layer Elements

Application elements are used to models components and systems that do the work required by other architectural concerns.

Table of Application Layer Elements

Application Component 	An *application component* represents an encapsulation of application functionality aligned to implementation structure, which is modular and replaceable.
Application Collaboration 	An *application collaboration* represents an aggregate of two or more application internal active structure elements that work together to perform collective application behavior.
Application Interface 	An *application interface* represents a point of access where application services are made available to a user, another application component, or a node.
Application Function 	An *application function* represents automated behavior that can be performed by an application component.
Application Interaction 	An *application interaction* represents a unit of collective application behavior performed by (a collaboration of) two or more application components.
Application Process 	An *application process* represents a sequence of application behaviors that achieves a specific result.
Application Event 	An *application event* represents an application state change.

Application Service 	An *application service* represents an explicitly defined exposed application behavior.
Data Object 	A *data object* represents data structured for automated processing.

Application Layer Example Diagrams

Application Components, Interfaces and Collaborations

This diagram demonstrates the use of Application Components, Application Interfaces and Application Collaborations The Online Travel Insurance Sales Collaboration is aggregated of a Quotation and Purchase system modeled as Application Components. The Collaboration is composed of a Web Service Interface which in turn Serves the Travel Website.

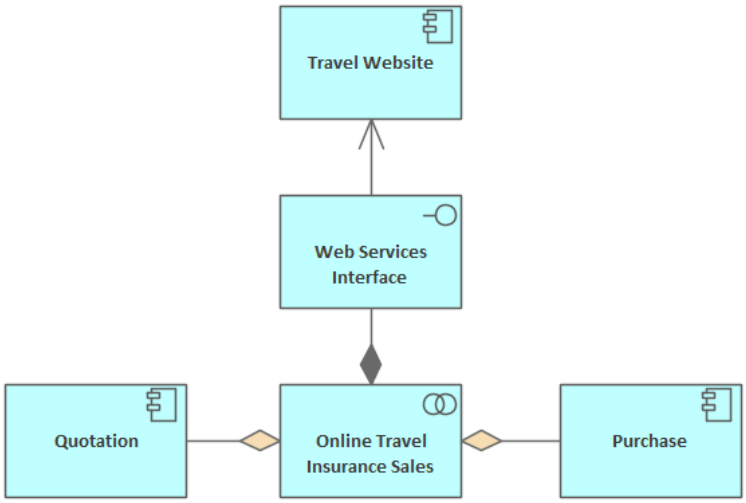
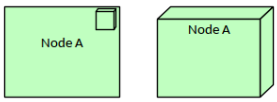
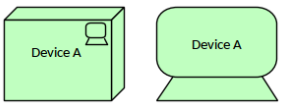
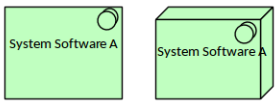
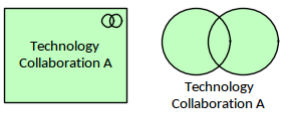
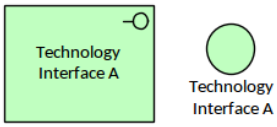
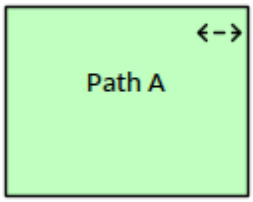


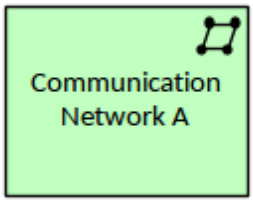
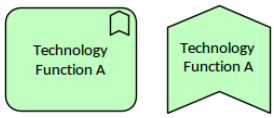
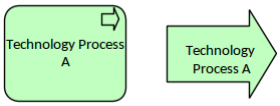
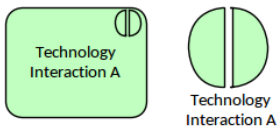
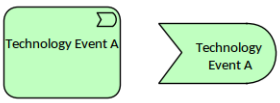
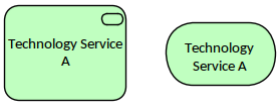
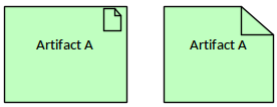
Figure: Example diagram from the ArchiMate Specification showing Application Components, Interfaces and Collaborations

Technology Layer Elements

Technology elements are typically used to model computer resources such as computers, servers and Networks.

Table of Technology Layer Elements

Node 	A *node* represents a computational or physical resource that hosts, manipulates, or interacts with other computational or physical resources.
Device 	A *device* represents a physical IT resource upon which system software and artifacts can be stored or deployed for execution.
System Software 	*System software* represents software that provides or contributes to an environment for storing, executing, and using software or data deployed within it.
Technology Collaboration 	A *technology collaboration* represents an aggregate of two or more technology internal active structure elements that work together to perform collective technology behavior.
Technology Interface 	A *technology interface* represents a point of access where technology services offered by a node can be accessed.
Path 	A *path* represents a link between two or more nodes, through which these nodes can exchange data, energy, or material.
Communication Network	A *communication network* represents a set of structures that connects nodes for transmission, routing, and reception of data.

	
Technology Function 	A *technology function* represents a collection of technology behavior that can be performed by a node.
Technology Process 	A *technology process* represents a sequence of technology behaviors that achieves a specific result.
Technology Interaction 	A *technology interaction* represents a unit of collective technology behavior performed by (a collaboration of) two or more nodes.
Technology Event 	A *technology event* represents a technology state change.
Technology Service 	A *technology service* represents an explicitly defined exposed technology behavior.
Artifact 	An *artifact* represents a piece of data that is used or produced in a software development process, or by deployment and operation of an IT system.

Technology Examples

Technology Functions, Interfaces, Processes and Events

This diagrams concerns Technology Functions, Interfaces, Processes and Events. The Technology Architect has modeled the Replicate Remote Data Process and the Technology Services and Functions that are required to support the Service. The Database Replication Function is composed of four other Functions that are represented using the 'nested' visualization option. The Composition relationships still exist between each of the four Technology Functions and the Database Replication function they are just suppressed in the diagram.

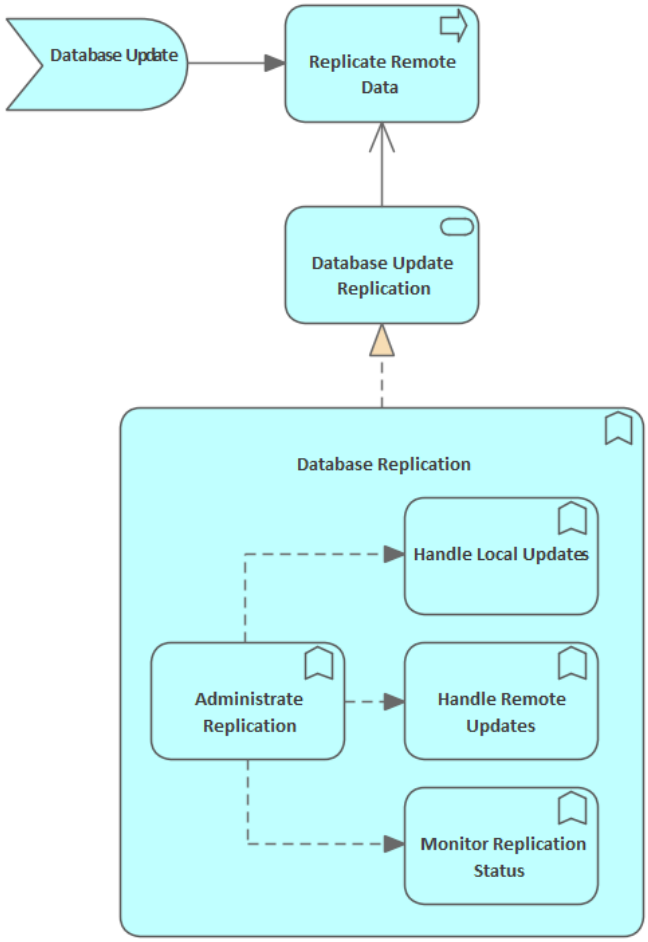
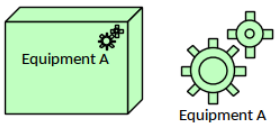
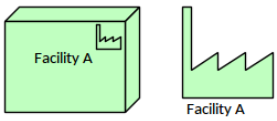
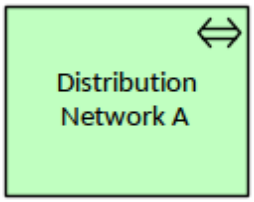
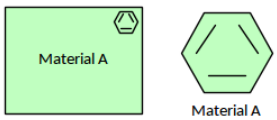


Figure: Example diagram from the ArchiMate Specification showing Technology Functions, Interfaces, Processes and Events.

Table of Physical Layer Elements

Physical elements are used to model 'things' in the real world such as facilities, equipment and logistic infrastructure.

Table of Physical Elements

Equipment 	*Equipment* represents one or more physical machines, tools, or instruments that can create, use, store, move, or transform materials.
Facility 	A *Facility* represents a physical structure or environment.
Distribution Network 	A *Distribution Network* represents a physical network used to transport materials or energy.
Material 	*Material* represents tangible physical matter or energy.

Physical Examples

This diagram is concerned with Physical things that are important to the Architecture, including Facilities, Equipment, Distribution Networks and Material. We see two other elements that the modeler has used; one is Path from the Technology Layer and the other is Location, which is a Composite element. The diagram describes what is being produced at a Manufacturing Plant modeled as a Facility, and the logistics in the form of Shipping of overseas items and trucking for domestic items to move the assembled items to local and national Distribution Centers.

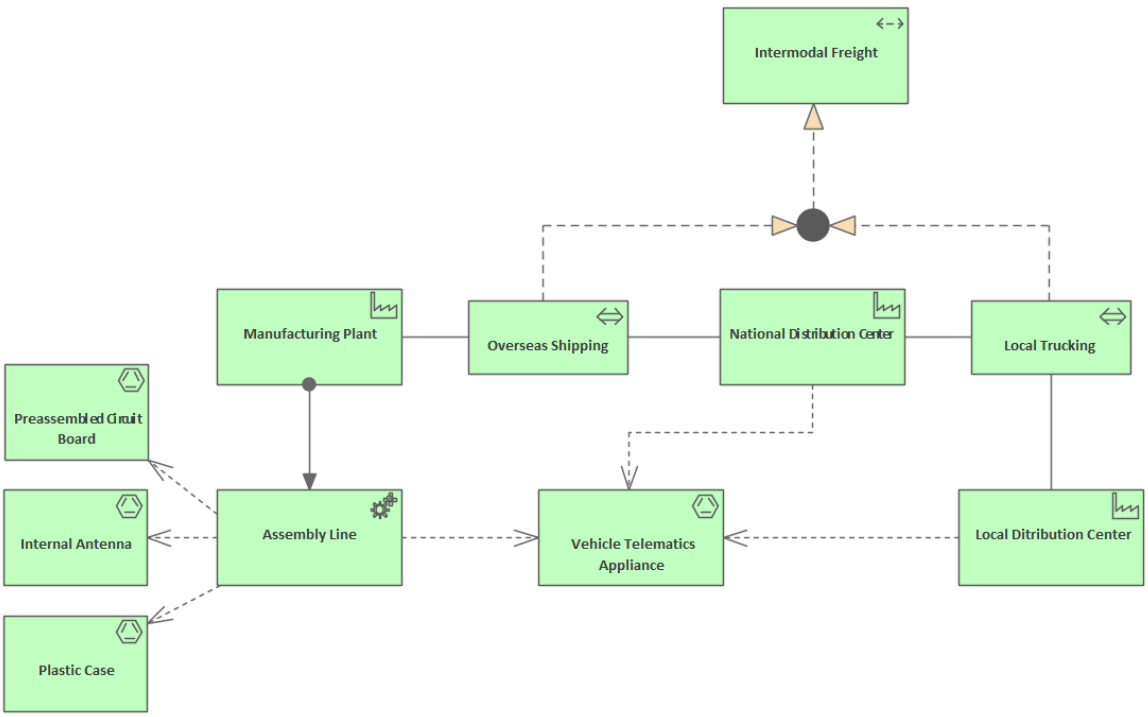
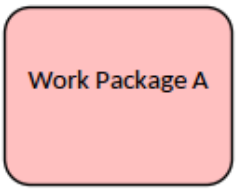
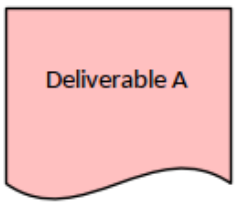
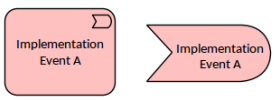
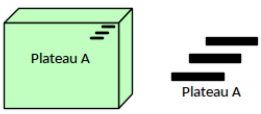
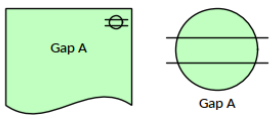


Figure: Example diagram from the ArchiMate Specification showing Facilities, Equipment, Material and Distribution Networks.

Table of Implementation and Migration Layer Elements

Implementation and Migration elements are used to model project related items such as work pages and deliverables.

Table of Implementation and Migration Elements

Work Package 	A *Work Package* represents a series of actions identified and designed to achieve specific results within specified time and resource constraints.
Deliverable 	A *Deliverable* represents a precisely-defined result of a Work Package.
Implementation Event 	An *Implementation Event* represents a state change related to implementation or migration.
Plateau 	A *Plateau* represents a relatively stable state of the architecture that exists during a limited period of time.
Gap 	A *Gap* represents a statement of difference between two plateaus.

Implementation and Migration Example

Plateaus, Gaps, Deliverables, Work Packages and Events

This diagram shows a number of Plateaus and the Gaps that are Realized by a number of Deliverables, which in turn are realized by a number of Work Packages. In this example the Work Package 'Next Generation Services Program' is composed of three subordinate Work Packages. The first of these - 'Architecture and Planning' - is triggered when the Program Approved Event is fired. This Work Package realizes a series of Deliverables, which in turn realize the Plateau. Gaps are used to record the result of Gap Analysis workshops and are important inputs for migration planning.

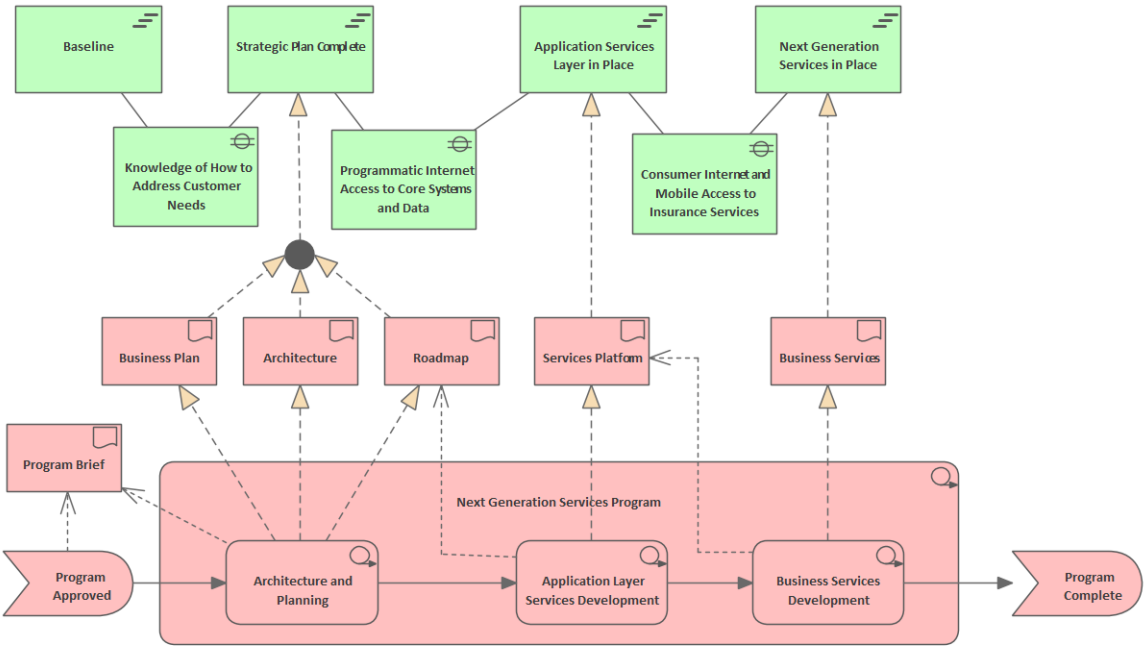
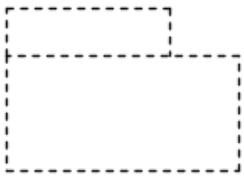
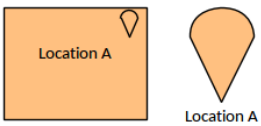


Figure: Example diagram from the ArchiMate Specification, showing Plateaus, Gaps, Deliverables, Work Packages and Events.

Composite Elements

Composite elements are special elements that consist of other concepts, from one or - more typically - multiple aspects or layers of the ArchiMate language. There are two composite elements, as described in this table. Composite elements can themselves aggregate or compose other composite elements, leading to a hierarchy of these elements.

Table of Composite Elements

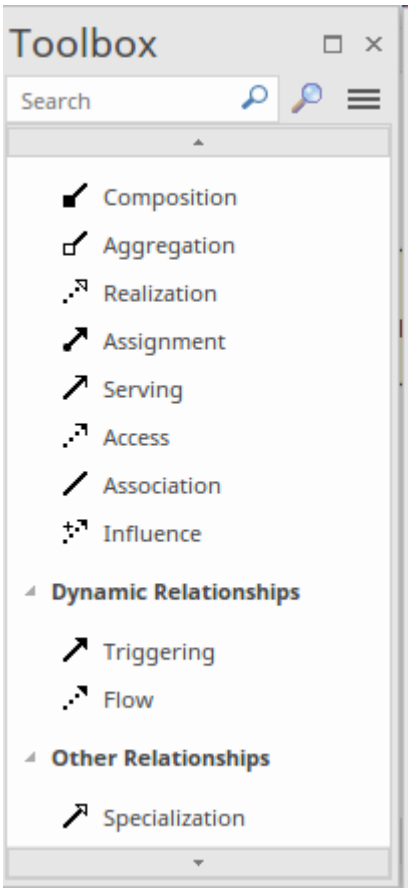
Grouping 	The *Grouping* element aggregates or composes concepts that belong together based on some common characteristic.
Location 	A *Location* represents a conceptual or physical place or position where concepts are located (e.g. structure elements) or performed (e.g. behavior elements).

Relationships

Relationships are an important part of the ArchiMate language; they bind elements to other elements and - in some cases - other relationships. Enterprise Architects allows relationships to be created in a number of ways; in this section we will discuss three of the main ways of creating relationships.

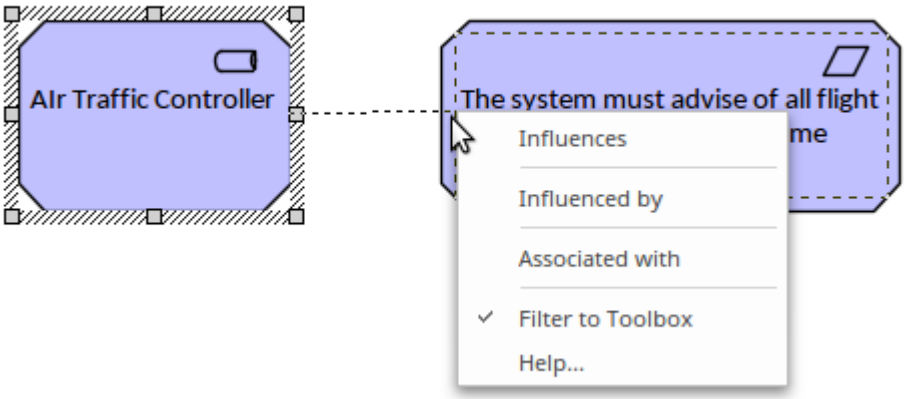
Creating Relationships - Toolbox

Relationships can be selected from the diagrams toolbox pages which are specific to the diagram or view that is being constructed. The relationship appear in a toolbox page beneath the elements and can be dragged between any two elements or in some cases between an element and a relationship. The relationships can be used with any pair of elements in the repository.



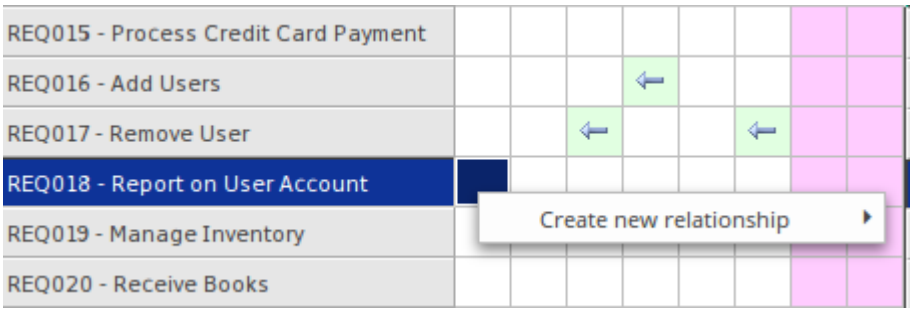
Creating Relationships - Quick Linker

Relationships can be drawn between two elements or an element and another relationship directly in a diagram, using the element or relationship Quick Linker. The drawing of a relationship is initiated by dragging the small vertical arrow that is positioned at the top right of an element or in the center of a relationship. The allowable elements will be displayed when the linker is drawn to target a diagram object.



Creating Relationships - Matrix

Relationships can be created using the matrix, which is a great tool for visualizing the relationship between two sets of elements in a spreadsheet-like appearance. The matrix displays the two sets of element based on the Package location and their type. The type of relationship can be specified, which then displays an arrow that indicates pairs of elements - one from the x-axis and one from the y-axis are related by the specified relationship. A new relationship can be added by right-clicking in an empty cell and selecting 'Create new relationship'.

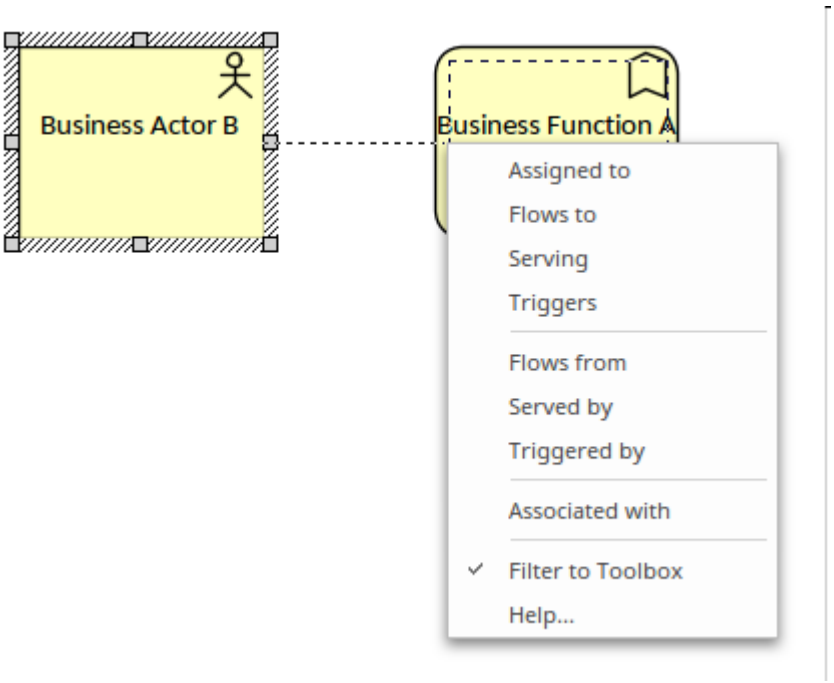


Relationships Overview

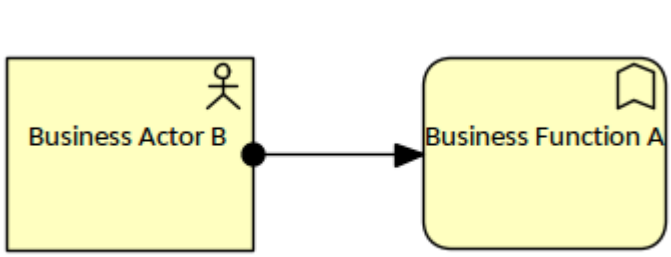
Enterprise Architect is a practical language-compliant platform that makes it easy to create industry standard Enterprise Architecture diagrams using ArchiMate. The tool allows the modeler to work with relationships flexibly, including being able to change the styles of the connectors while at the same time maintaining the rules and constraints of the ArchiMate language, including restricting what elements can be connected together.

Allowable Element Relationships

The compliance has been achieved by building the ArchiMate facility on top of a flexible and compliant meta-model that implements the relationship table specified in Appendix B of the ArchiMate Specification. This ensures that if the specification changes it is simply a matter of updating the metamodel and all aspects of the tool will reflect the change. This makes it easy for the user to create first-class and compliant models as the interface restricts the available relationships when attempting to draw a connection between two elements in a diagram canvas.



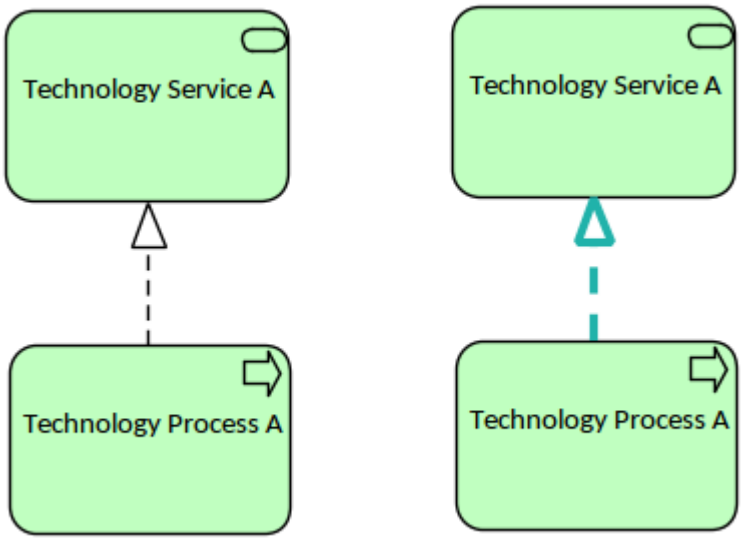
In this example we see a modeler using the Quick Linker (a fast way of connecting elements) to create a relationship from a Business Actor to a Business Function. Once the relationship is dropped onto the target a menu is opened that displays the allowed relationships between the two element types based on the direction of the connection.



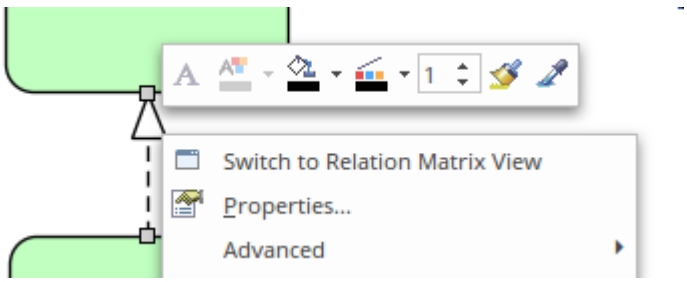
In this diagram we see that the user has selected the assignment connector, as this is an allowed relationship from a Business Actor to a Business Function.

Changing Relationship Presentation

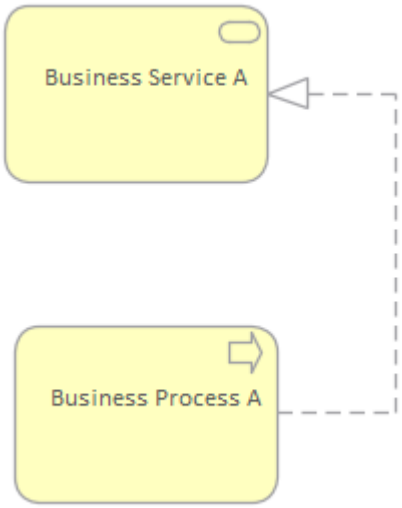
Enterprise Architect provides excellent flexibility for changing relationships, elements, diagram styles, and themes. An architect or other user can change any of the standard ArchiMate relationships to different sizes and colors without affecting the meaning or syntax of the diagrams. After a user has changed the presentation, any architect or user familiar with the language will recognize the graphical notation of the relationship. This illustration shows two diagrams showing the exact relationship in different line widths and colors.



The options to change the style of a relationship are available from a number of locations, including the ribbons and the element Quick Style icon that is displayed when the relationship's context menu is selected in the diagram. This illustration shows the options available from the diagram.



A line style can be applied to a any connector which will change the presentation to styles such as auto routing, lateral, orthogonal and tree vertical and horizontal styles. There is a custom style which you can use to apply any number of user-defined way-points.

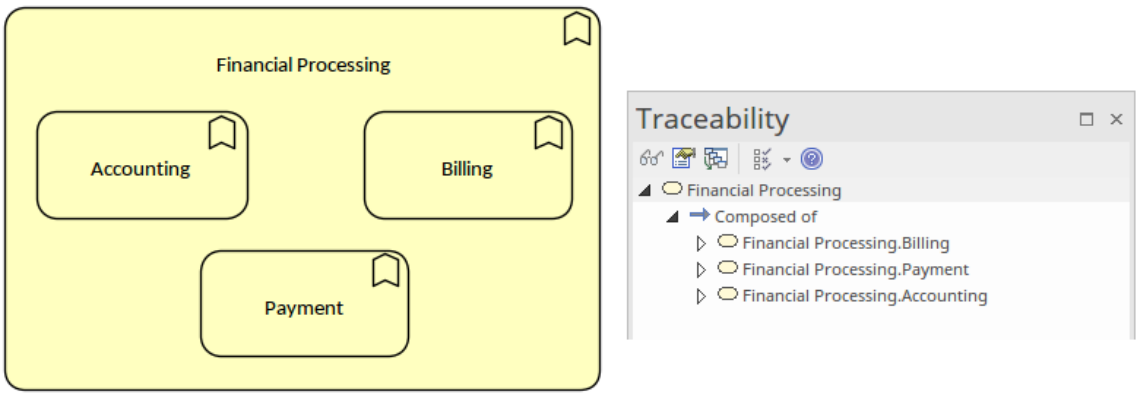


Relationships can be further customized using the profile system. Using a profile a user can customize name, attributes, symbols and colors and other aspects of the relationship. For further information see the Customizing the ArchiMate Language topic in this document.

Nesting

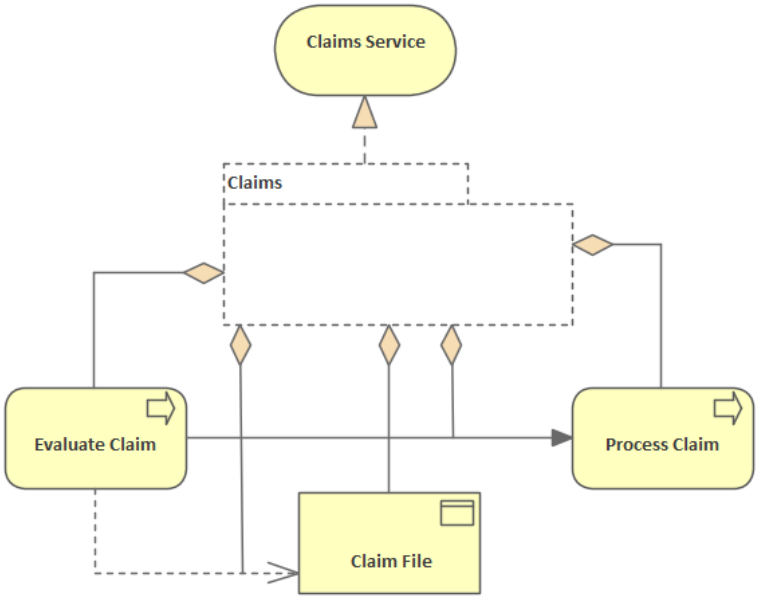
Enterprise Architect allows elements to be nested as an alternative way of representing relationships. This is a handy and compact notation for some diagrams, alleviating the issue of some stakeholders finding the ArchiMate visual syntax difficult to interpret. It does, however, introduce ambiguities when the diagrams are viewed outside the tool. This is an issue with the ArchiMate language itself rather than being any limitation of the tool, and while the diagrams are being viewed inside Enterprise Architect the ambiguities can always be resolved. Enterprise Architect allows nested views to be created and visualized, providing a number of ways to remove the ambiguities by use of other windows that can be opened in the workspace; these clearly indicate the type of relationship implied by the element nesting.

To create nested elements, the relationships between elements must already exist or first be created. A modeler can then use drag-and-drop to nest one or more elements inside another element. Enterprise Architect hides the relationships in the diagram but they are visible through a number of other workspace windows, including the Traceability window, the Relationships window and the 'Details' tab of the Inspector window. This illustration shows the Traceability window when the containing element is selected, clearly showing the Composition relationships that exist in the model; these are not displayed as direct relationships in the diagram but are indicated by the nesting.



Connecting to Other Relationships

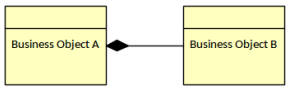
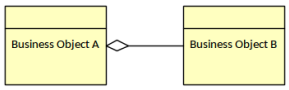
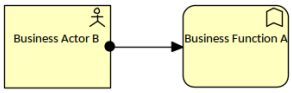
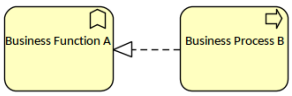
Enterprise Architect, in compliance with the ArchiMate specification, allows you to connect one end of certain relationships to other relationships. This is an effective device to express particular modeling concepts; for example, the data that is transmitted between two elements such as two Application Components or Business Functions. This diagram shows two examples where this feature has been used.



Structural Relationships

Structural relationships depict the 'static' coherence within an architecture. There is always an element at the uniting end (from end) of the relationship, for the assignment and realization relationships it can be an element or a relationships connector. The uniting end (to end) of the relationship typically connects to an element but in some circumstances it can be another relationship or relationship connector. These relationships are not only about how things are structured but also about responsibility for the performance of behavior, storage, or execution or the representation of how a more concrete element plays a critical role in the creation, achievement, sustenance or operation. This table describes the structural relationships.

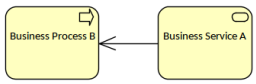
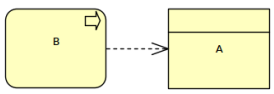
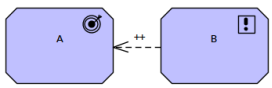
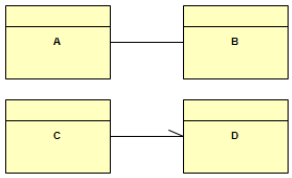
Structural Relationships

Composition 	The *composition* relationship represents that an element consists of one or more other concepts.
Aggregation 	The *aggregation* relationship represents that an element combines one or more other concepts.
Assignment 	The *assignment* relationship represents the allocation of responsibility, performance of behavior, storage, or execution.
Realization 	The *realization* relationship represents that an entity plays a critical role in the creation, achievement, sustenance, or operation of a more abstract entity.

Dependency Relationships

Dependency relationships describe how elements support or are used by other elements. They can indicated direction in the form of an arrowhead at one end of the line - the direction is optional for some relationships () and mandatory for others (Influence). Four types of dependency relationship are distinguished as indicated in this table.

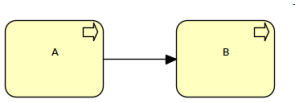
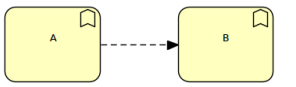
Dependency Relationships

Serving 	The *serving* relationship represents that an element provides its functionality to another element.
Access 	The *access* relationship represents the ability of behavior and active structure elements to observe or act upon passive structure elements.
Influence 	The *influence* relationship represents that an element affects the implementation or achievement of some motivation element.
Association 	An *association* relationship represents an unspecified relationship, or one that is not represented by another ArchiMate relationship.

Dynamic Relationships

The dynamic relationships describe temporal dependencies between elements within the architecture. They are often mistakenly interpreted as causal relationships implying that the element at the tail end of the relationship somehow causes the element at the arrow end to 'fire' - this is not correct. In the case of the Flow relationship an item can be transmitted between the two connected elements. Two types of dynamic relationship are distinguished as indicated in this table.

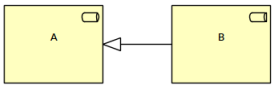
Dynamic Relationships

Triggering 	The *triggering* relationship represents a temporal or causal relationship between elements.
Flow 	The *flow* relationship represents transfer from one element to another.

Other Relationships

The other relationships category is a catch all label for relationships that don't fit into the other categories. There is only one relationship in this category entitled Specialization which is a directed relationship of family meaning that the element at the tail end is a more specific than the element at the arrow-end of the relationship.

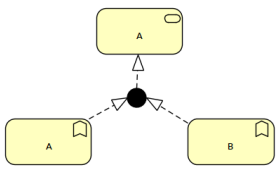
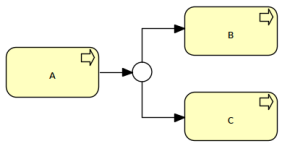
Other Relationships

Specialization 	The *specialization* relationship represents that an element is a particular kind of another element.

Relationship Connectors

A junction is used in a number of situations to connect relationships of the same type. A path with junctions that connect relationships of this type is only allowed between two concepts, if a direct relationship of that type between these concepts is also permitted. Simply put, you cannot use junctions to create relationships between concepts that would otherwise not be allowed.

Connectors

Junction (AND) 	A junction is used to connect relationships of the same type to indicate that all connectors on the multiple side(s) of the junction will be invoked.
Junction (OR) 	A junction is used to connect relationships of the same type to indicate that one (and only one) connector on the multiple side(s) of the junction will be invoked.

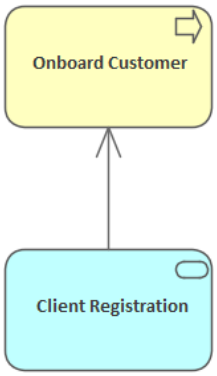
Alignment between Layers

The ArchiMate language is divided into a number of layers which is a useful mechanism to separate the concerns of various disciplines (e.g. Business from Application layers). This creates a division in the models that are created and so we need a mechanism to bridge this separation so that we can ensure that the elements and relationships created in one layer are able to be visualized in relation to the elements in the layer above or below (for example, elements in the business layer should be able to connect to elements in the strategic layer above and the application layer below).

ArchiMate has two important relationships that are useful for this purpose.

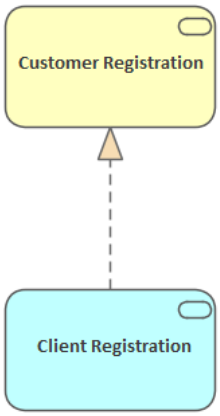
Serving Relationship

Serving relationships can be used to connect an application layer element to a business layer element for example between an application service and the different types of business behavior elements such as business processes and functions, and between application interfaces and business roles. The relationships can also be drawn in the opposite direction between business services and application behavior elements, and between business interface and application component. These relationships represent the behavioral and structural aspects of the support of the business by applications.



Realization Relationship

Realization relationships can be used to connect an application layer element to a business layer element for example an application process or function can be said to realize a business process or function, or a data object or a technology object can be said to realize a business object, so as to indicate that the data object is a digital representation of the corresponding business object, or the technology object is a physical representation of the business object. The relationship is always drawn in the direction from the lower layer with the arrow pointing to the higher layer.


Customer Registration
Client Registration

Customizing the ArchiMate Language

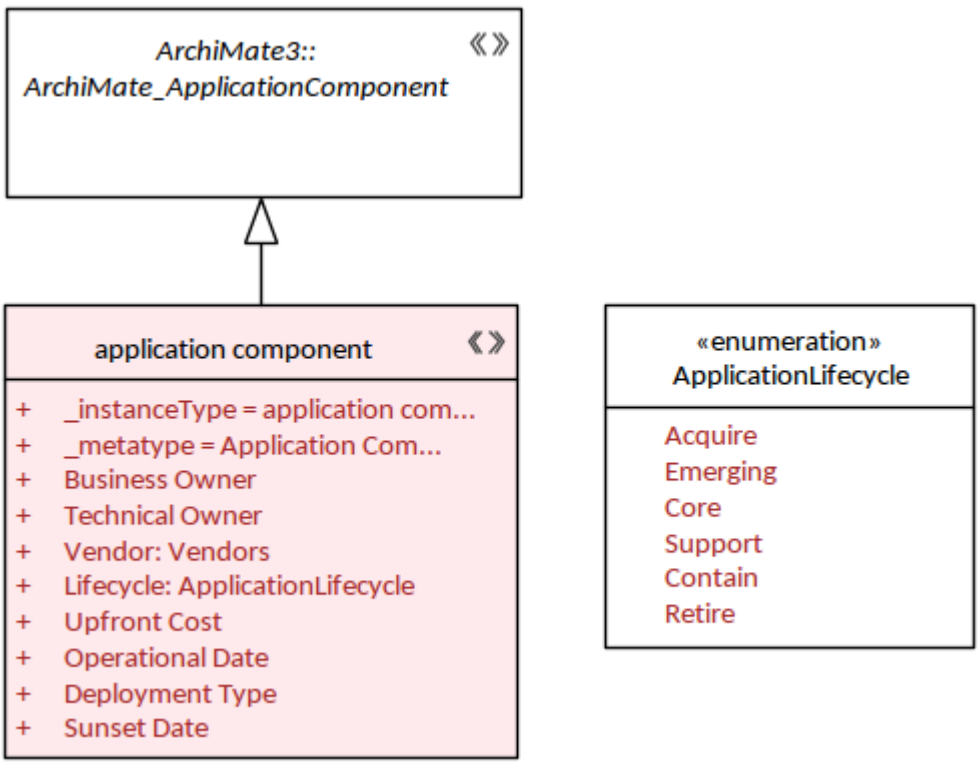
Enterprise Architect provides a handy capability to customize language using its profile system, which is part of a sophisticated feature to create Model Driven Generation extensions to the tool. These extensions can be designed and built within the tool and the resulting technology can be imported into the tool making customizations available to Enterprise Architects and other repository users.

A typical usage scenario would be that a team might want to categorize their Application Portfolio based on a range of criteria and be able to analyze architectures based on the applications to answer questions like:

- What are the set of technologies underpinning our mission critical systems
- Which applications run on a Cloud platform
- Which Business Capabilities would be compromised if Vendor A became insolvent

This would require the provision and maintenance of properties for each application. This can be done using the profile system where any number of properties can be added to an ArchiMate element or connector and the customized element can be added to custom Toolbox pages. The elements and connectors are then available to the architect for modeling and once the properties are populated with organization specific data, reports and visualizations can be created that use the properties.

In this diagram we see the customization of an ArchiMate Application Component as part of the creation of a profile. We see that the user-defined *Application Component* is a specialization of the standard ArchiMate element (indicated by the triangular arrow-headed relationship). We also see that the specialized element has a number of added properties such as Business Owner, Technical Owner and Deployment Type. These are user-defined properties and provide the mechanism for adding organization-specific information that will help in the analysis of an architecture to answer questions like those presented earlier.



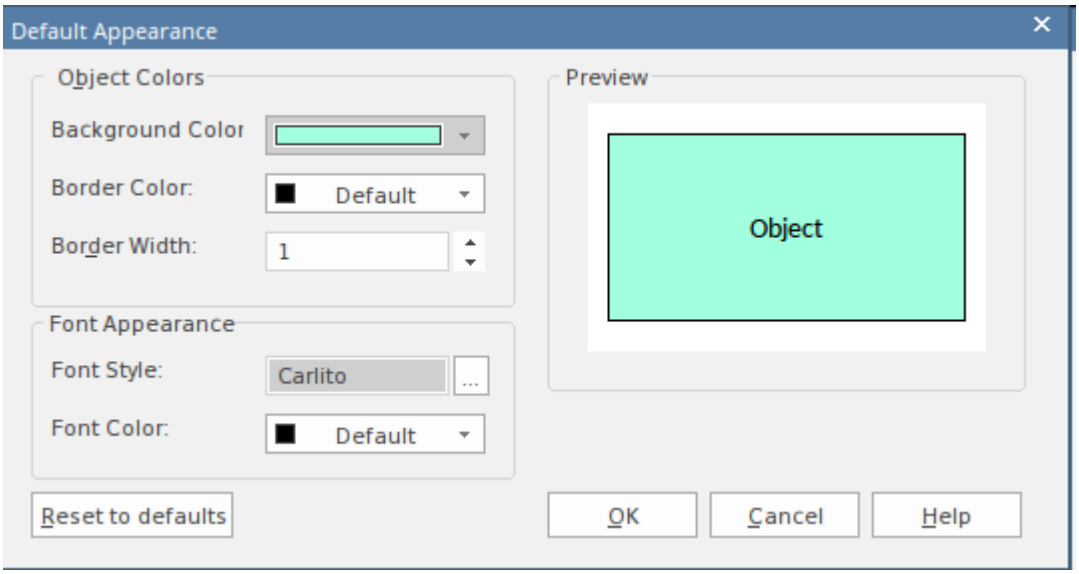
Custom Views and ViewPoints

Enterprise Architect has a wide range of built-in diagram views, but you can also create your own Metamodels that define custom diagram Views. For example, you might define a specific Metamodel that addresses the needs of Security Architecture modeling in your organization, and then mandate that all Security Views use that diagram View. You can

quickly add your diagram Views to the current model, where you or other modelers can apply them to your diagrams.

Changing Element Default Appearance

Enterprise Architect allows the appearance of elements in diagrams to be altered to suit the intended audience. When creating a profile and a customized element it is also possible to set its default appearance so when it is added to any diagram it will take on that appearance. For example you might want to use a different set of colors that has meaning in your architecture practice. This illustration shows the mechanism for setting the default color, font and other presentation options.



Changing an Element or Connector's Shape

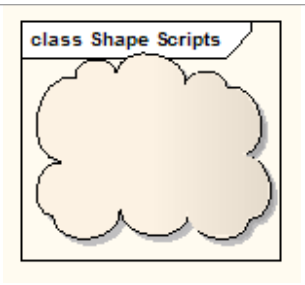
While the standard ArchiMate element shapes are important for communicating to audiences, there are situations where you might want to change the shape of an element that you have customized as part of a profile. For example, an organization is a vendor of 'Smart Phones' and wants to represent a product with an icon resembling the phone. Alternatively, an architect wants to represent a network using a Cloud as an alternative to a standard Technology element icon.

Shape Script Example

You can create a wide range of shapes, effects and text statements using Shape Scripts, to enhance the appearance and information value of the elements and connectors you create. An example of a scripts is provided in this table.

Shape	Example Script

	// CLOUD PATH EXAMPLE SHAPE
	shape main
	{
	StartCloudPath();
	Rectangle(0, 0, 100, 100);

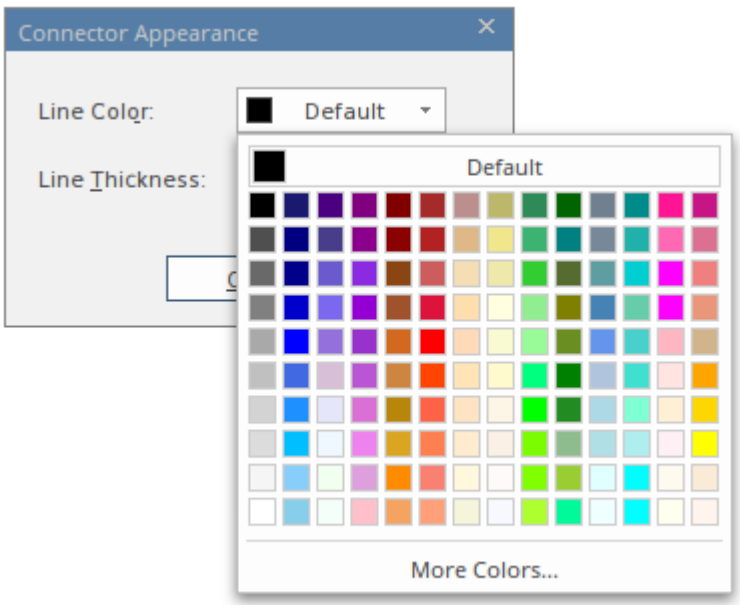



```
    EndPath();
    FillAndStrokePath();
}
```

Changing Connector Default Appearance

Enterprise Architect allows the appearance of connectors in diagrams to be altered to suit the intended audience. When creating a profile and a customized connector it is also possible to set its default appearance so when it is added to any diagram it will take on that appearance. For example you might want to use a different set of colors that has meaning in your architecture practice. This illustration shows the mechanism for setting the default color, font and other presentation options.

-



Creating Proprietary Elements and Relationships

In addition to customizing existing ArchiMate elements as described earlier, it is possible to add completely new and proprietary elements and connectors, as long as they don't have any conflicting or obstructing meaning with respect to the corpus of ArchiMate elements and customizations. An example of this is in the Security Architecture Domain, where an architect might want to have an element that represents a Security Policy.

Example Profile

Enterprise Architect is a highly compliant tool for modeling enterprise architectures using the ArchiMate language; in compliance with the specification it allows users, teams, organizations or industries to customize the language. This is always intended as a specialization of the existing language constructs and is never intended to compromise or compete with the linguistic structure of the language. This is a very useful feature and provides a mechanism for the addition of user-defined properties to elements, relationships and relationship connectors. For example engineering organizations might want to add user defined properties associated with safety or performance whereas a financial organization might want to add properties concerning money or credit.

This example shows the specialization of an ArchiMate business actor into two elements and an ArchiMate Flow relationship into a Money Flow. The Organizational Unit, for example, has two user defined properties added, namely:

- Corporate Level
- Location

The specialized relationship Money Flow, for example, has two user defined properties added, namely:

- Amount
- Currency

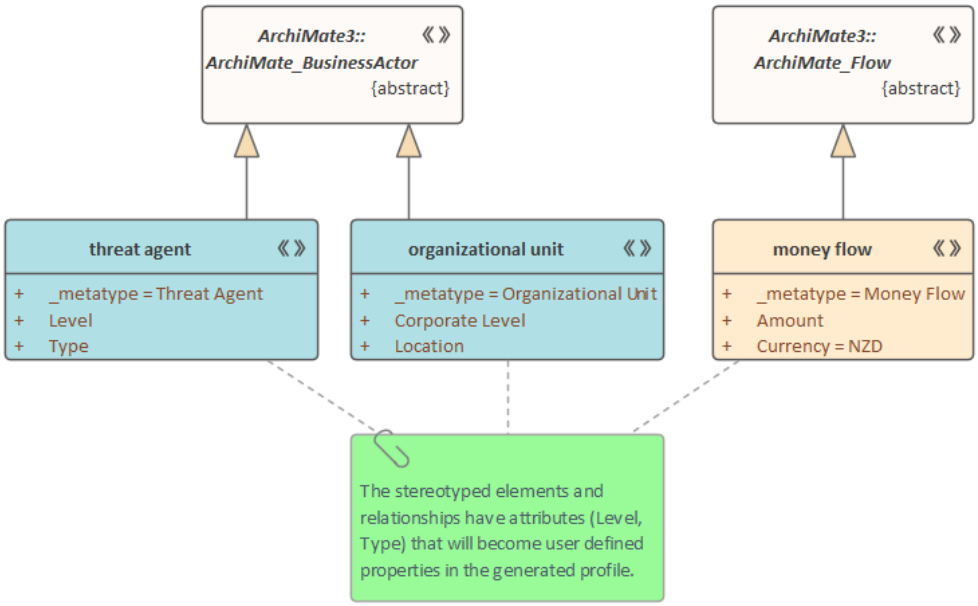


Figure: Showing the specialization of an ArchiMate business actor and a flow relationship

Architecture Views and Viewpoints

The Architecture Views and Viewpoints are a useful mechanism to provide stakeholders with relevant and meaningful visualization of the Enterprise Architecture that has been created in the repository. Every stakeholder - or, more commonly, a stakeholder group - will typically have different concerns and interests and so the View and Viewpoint mechanism provides a robust way of ensuring that they get value from the work that has been done to articulate the Enterprise Architecture. Essentially, a View is what you see when 'looking' from a particular standpoint (Viewpoint).

Overview of Views and Viewpoints

The tool has built-in support in a number of different places for the Viewpoint mechanism. Two of the most important places are the:

- Model Wizard Views and Viewpoints for ArchiMate
- Diagram Views available from the Diagram window

Creating Views Using the Toolbox

Enterprise Architect also supports creating Views from a blank canvas using the Diagram Viewpoint mechanism, which allows you to select an example Viewpoint as the basis for a new diagram. Choosing any of the available Viewpoints results in a new diagram being created, and a Toolbox being displayed in which the Viewpoint page only displays the available elements for that Viewpoint.

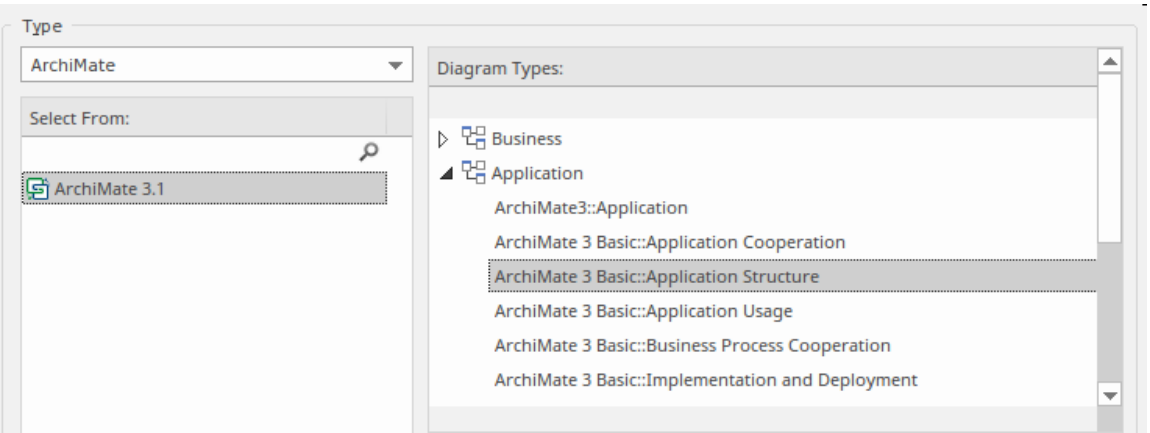


Figure: Shows the views available from the New Diagram dialog.

In the next diagram we see two Toolbox pages that are applicable for the Application Usage Viewpoint. The first page contains a subset of the Application Layer elements used for the Viewpoint and the second page contains a subset of the Business Layer elements.

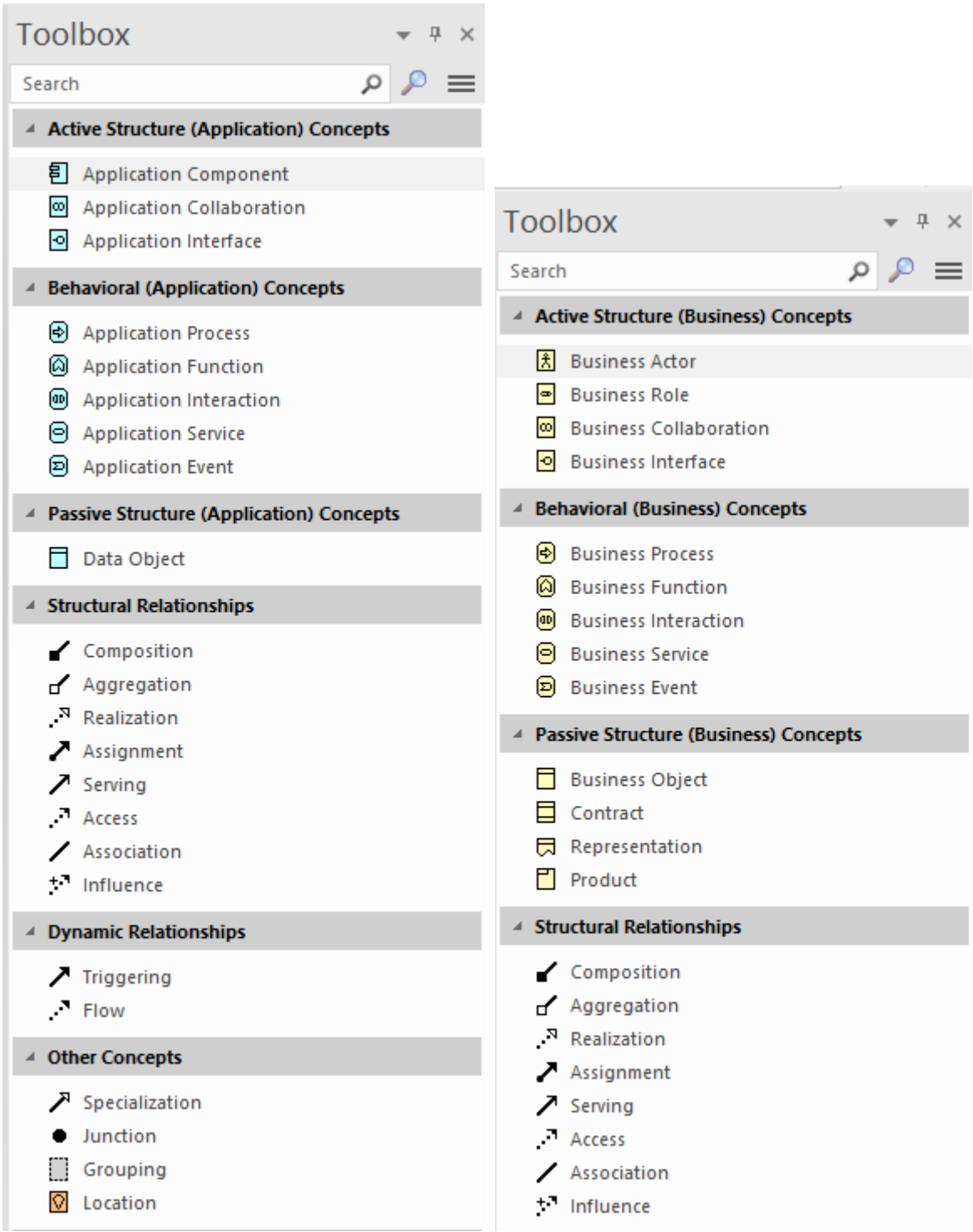


Figure: Showing the toolbox pages that list the available elements and relationships for the selected viewpoint.

Elements can be dragged and dropped from the Toolbox pages to the open diagram, which will create a new element and add it to the open diagram. Applicable elements can also be dragged onto the diagram from the Browser window, which allows existing elements that have been created or used in other diagrams to be added to the current diagram. For example, an Application Service and a Business Process might have been created at an earlier time and these could be added to a diagram that contains a new Application Component added to an Application Usage Viewpoint.

In this diagram we see that a modeler outside the Enterprise Architect group has created a UML Realization relationship from a UML Component in the Implementation model to an ArchiMate Component, thus linking the two different disciplines without breaking any ArchiMate language syntax rules.

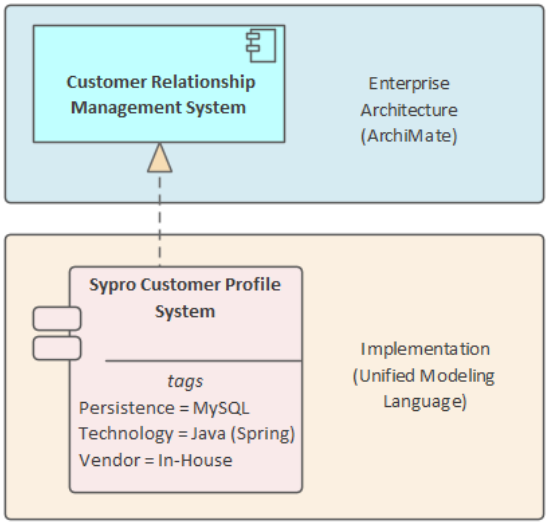
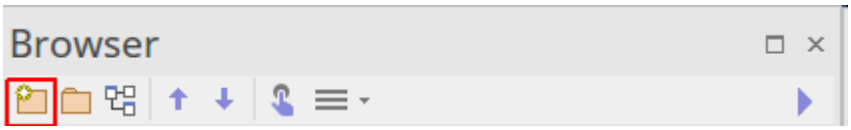


Figure: Showing how ArchiMate elements can be linked to model elements

Creating Views Using the Wizard

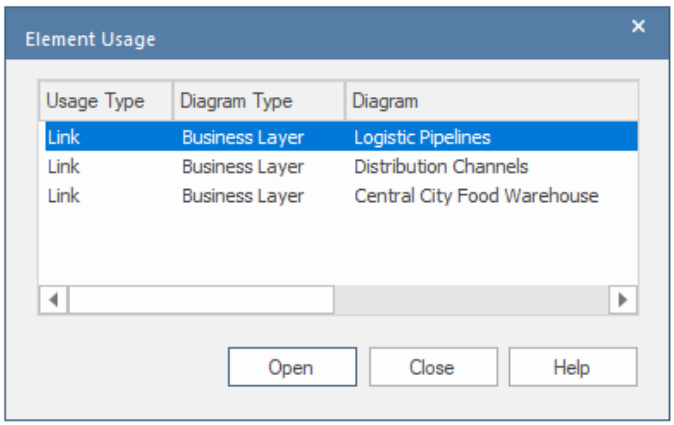
Enterprise Architect has full support for views and viewpoints, providing rich Model Wizard patterns where each viewpoint has been conveniently created and can be injected into the repository at any location. This image shows the icon to select in the Browser window toolbar to open the Model Wizard.



Tracking Elements Between Views

Enterprise has a number of windows that allow elements to be tracked between views and more broadly between any diagrams that contain the elements. This can be achieved by using the element's context menu and selecting 'Find in all Diagrams'. When selected, and when an elements does appear in multiple views, a window will provide a list of all the views that contain the element.

Central City Food Warehouse
Dry Storage
Wet Storage
Fresh Storage
Frozen Storage


Element Usage
Usage Type
Diagram Type
Diagram
Link
Business Layer
Logistic Pipelines
Link
Business Layer
Distribution Channels
Link
Business Layer
Central City Food Warehouse
Open
Close
Help

Basic Viewpoints

The basic Viewpoints are a reference or starting point for architectural descriptions and for creating meaningful and useful models. They act as a catalyst for an architect, allowing that person to get some modeled content down, thus removing the 'canvas-fright' syndrome that tends to paralyze progress. They are useful in supporting peer reviews, fostering organizational standards and helping novice modelers or newcomers to a domain. The ArchiMate specification has included all the Viewpoints as examples and it is well to remember that they should be tailored to suit the stakeholders who will ultimately need to digest the material. The basic Viewpoints are primarily intended for the architecture community. Some Viewpoints are limited to a single layer while others can contain elements from multiple layers.

Table of Basic Viewpoints

This table lists the example Basic Viewpoints.

Name	Description
Organization	Structure of the enterprise in terms of roles, departments, etc.
Application Structure	Shows the structure of a typical application in terms of its constituents.
Information Structure	Shows the structure of the information used in the enterprise.
Technology	Infrastructure and platforms underlying the enterprise’s information systems in terms of networks, devices, and system software.
Layered	Provides overview of architecture(s).
Physical	Physical environment and how this relates to IT infrastructure.
Product	Shows the contents of products.
Application Usage	Relates applications to their use in, for example, business processes.
Technology Usage	Shows how technology is used by applications.
Business Process Cooperation	Shows the relationships between various business processes.
Application Cooperation	Shows application components and their mutual relationships.
Service Realization	Shows how services are realized by the requisite behavior.
Implementation and Deployment	Shows how applications are mapped onto the underlying technology.

Organization Viewpoint

The organization viewpoint focuses on the (internal) organization of a company, department, network of companies, or other organizational entity. It is possible to present models in this viewpoint as nested Block diagrams, but also in a more traditional way, such as organizational charts. The organization viewpoint is very useful in identifying competencies, authority and responsibilities in an organization.

This image shows a diagram created from the Viewpoint Pattern, which is a built-in part of the Model Pattern facility.

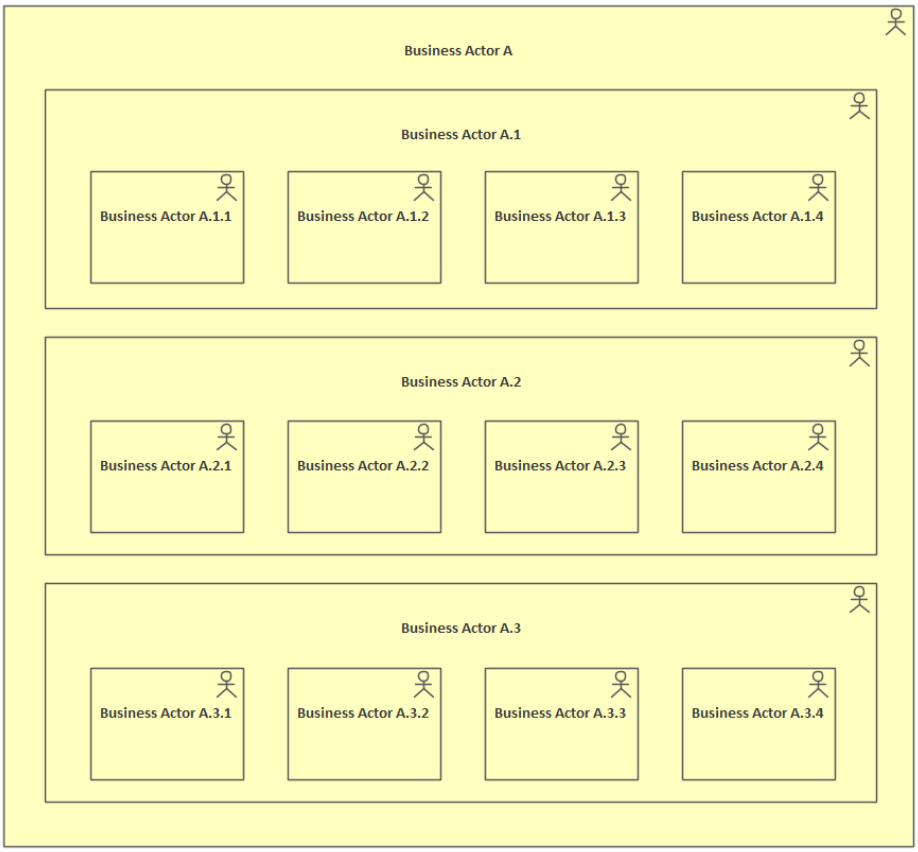


Figure: Showing the *Organization* viewpoint

Application Structure Viewpoint

The application structure viewpoint shows the structure of one or more applications or components. This viewpoint is useful in designing or understanding the main structure of applications or components and the associated data; for example, breaking down the structure of the system under construction, or identifying legacy application components that are suitable for migration/integration.

This image shows a diagram created from the Viewpoint Pattern, which is a built-in part of the Model Pattern facility.

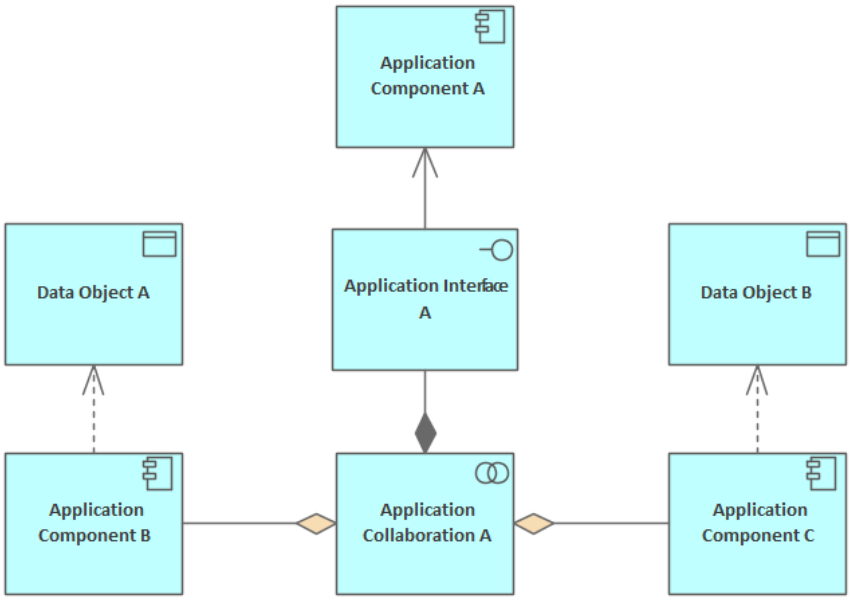


Figure: Showing the *Application Structure* viewpoint

Information Structure Viewpoint

The *Information Structure Viewpoint* pattern creates elements that show the structure of the information used in the enterprise or in a specific business process or application, in terms of data types or information elements. It will assist in visualizing the information from the business level through the application level down to the infrastructure elements that implement databases and other persistent stores.

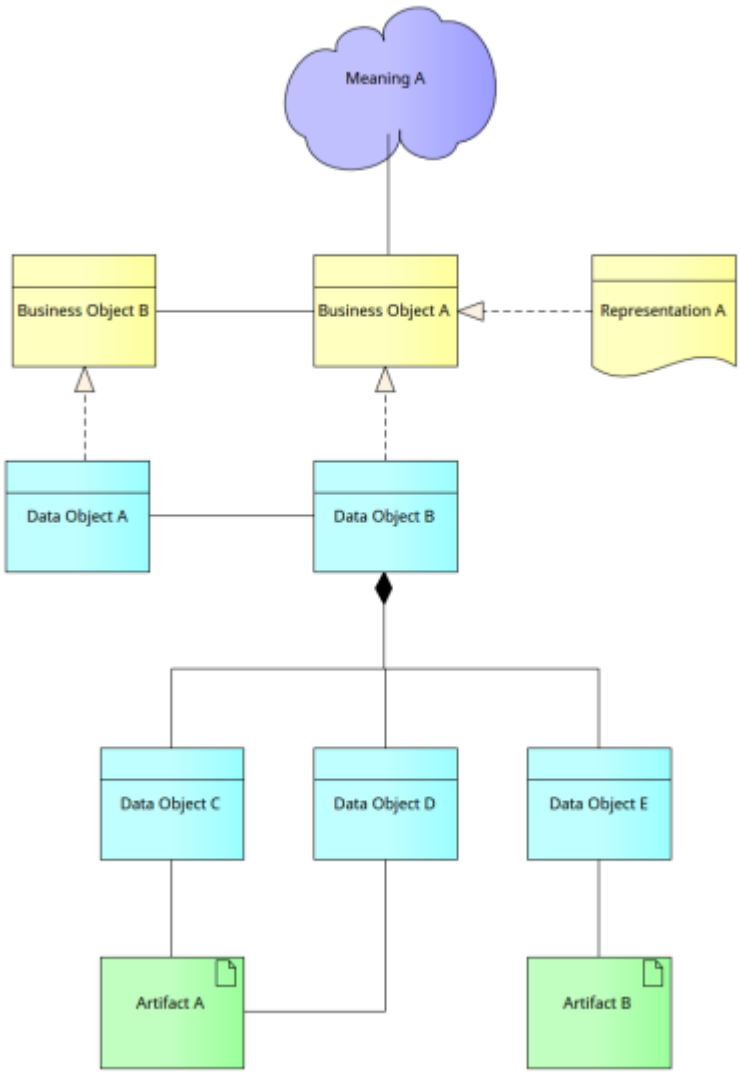


Figure: Showing the *Information Structure Viewpoint* viewpoint

Technology Viewpoint

The *Technology Viewpoint* pattern creates elements and a diagram that describes the software and hardware technology elements supporting the Application Layer, such as physical devices, networks, or system software such as middleware operating systems, databases and other containers.

Scope: Technology layer - Multiple aspects

This image shows a diagram created from the Viewpoint Pattern that is a built-in part of the Model Pattern facility.

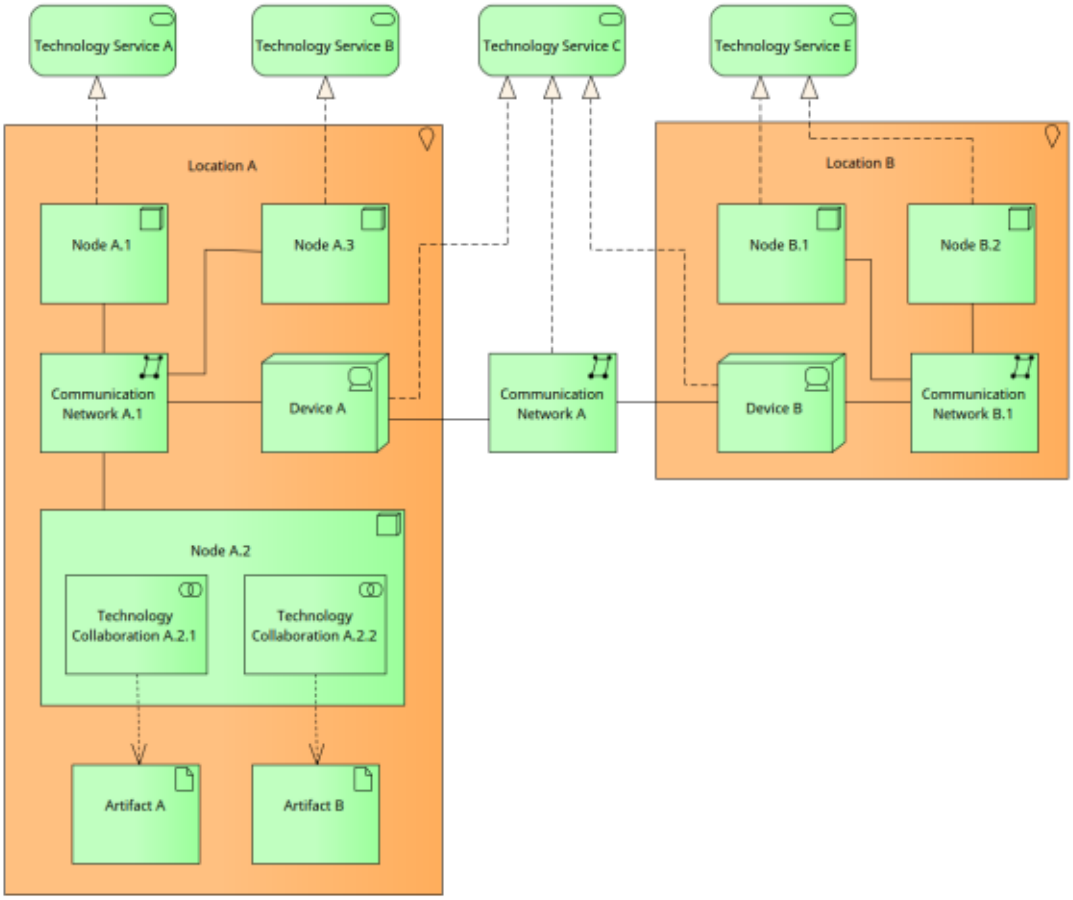


Figure: Showing the *Technology* Viewpoint

Layered Viewpoint

The *Layered Viewpoint* pattern creates a number of elements and diagrams that allow the visualization of multiple layers of an Enterprise Architecture in a single diagram. The partitioning which uses the Grouping element allows the representation of elements such as Business Processes in dedicated layers and elements such as Application Services in services layers. Any number of layers can be included but the diagram is most expressive when dedicated and service layers are interleaved.

This image shows a diagram created from the Viewpoint Pattern that is a built-in part of the Model Pattern facility.

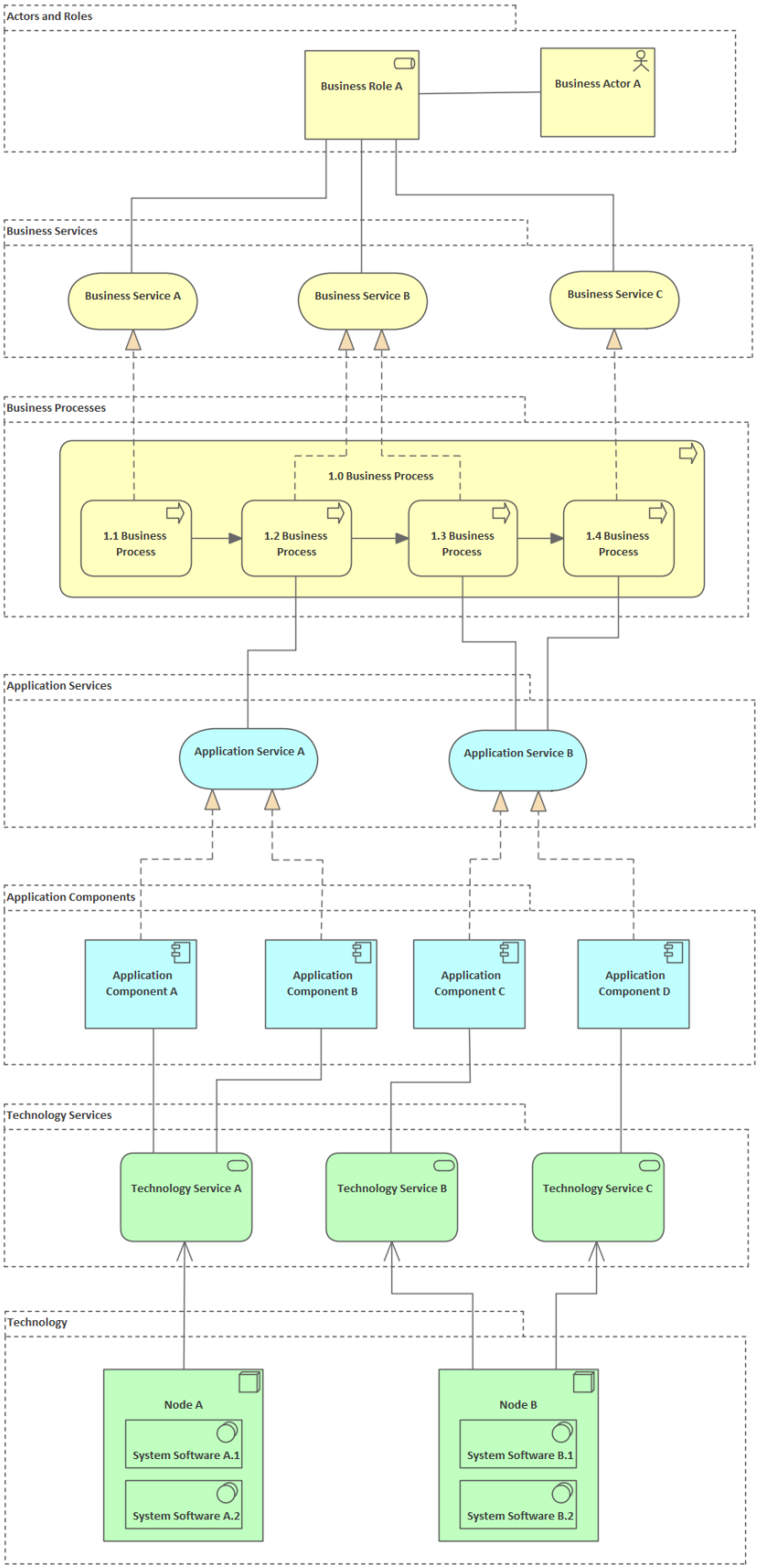


Figure: Showing the *Layered* Viewpoint

Physical Viewpoint

The *Physical Viewpoint* pattern creates elements and diagrams that contains equipment (one or more physical machines, tools, or instruments) that can create, use, store, move, or transform materials. It also describes how the equipment is connected via the distribution network and allows the visualization of other active elements that are assigned to the equipment.

-

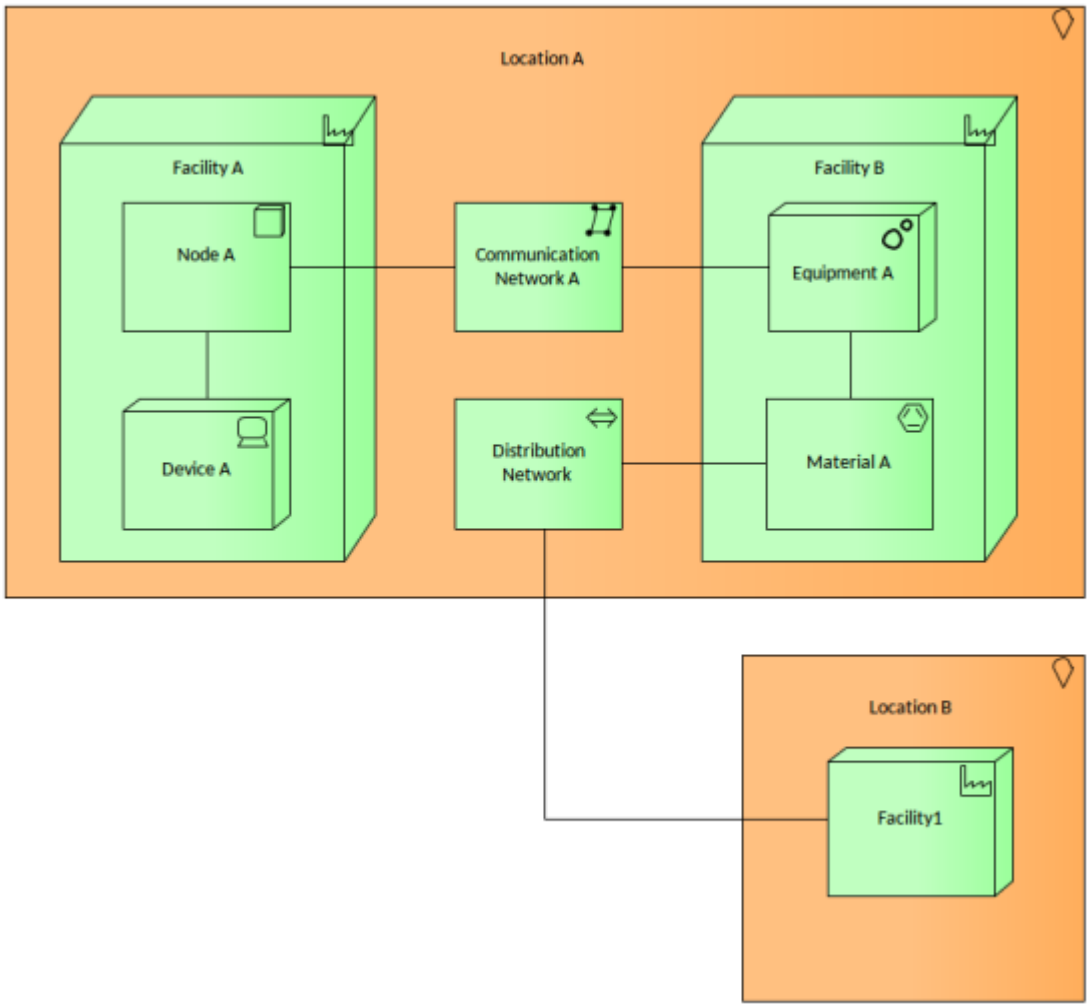


Figure: Showing the *Physical* Viewpoint

Product Viewpoint

The *Product Viewpoint* pattern creates elements and a diagram that describe the value that the products offer to external parties such as customers or other stakeholders It allows them to visualize one or more products' composition in terms of their constituent business, application, or technology services and any number of contracts or other agreements. The channels (interfaces) through which this product is offered, and the events associated with the product can also be represented in this viewpoint.

-

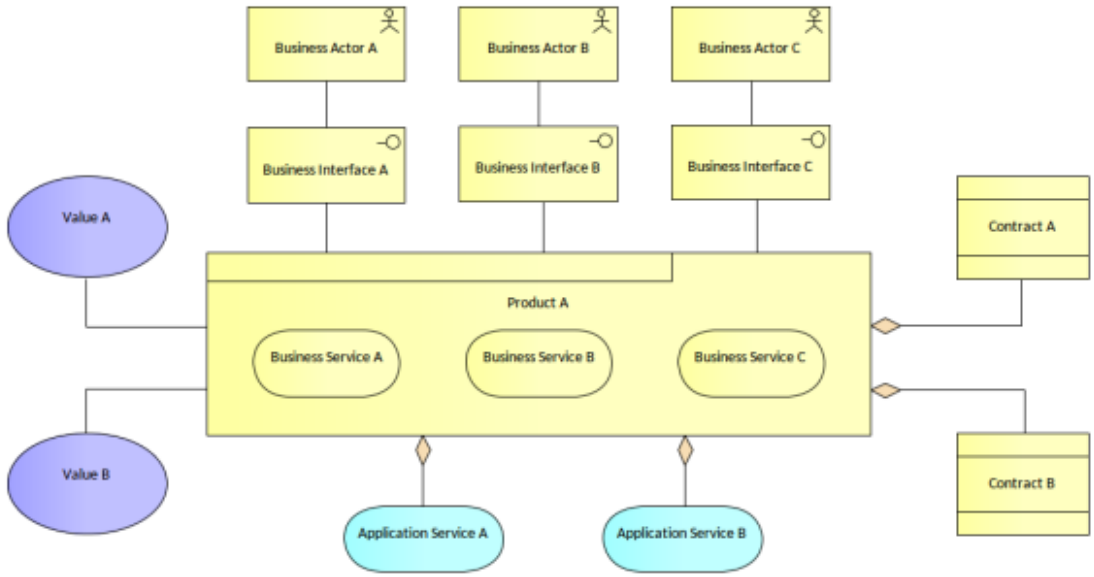


Figure: Showing the *Physical* Viewpoint

Application Usage Viewpoint

The *Application Usage Viewpoint* pattern creates elements and a diagram that describes how application services and the applications that realize them are used to support any number of business processes. It can also show the relationship between that applications that implement the services.

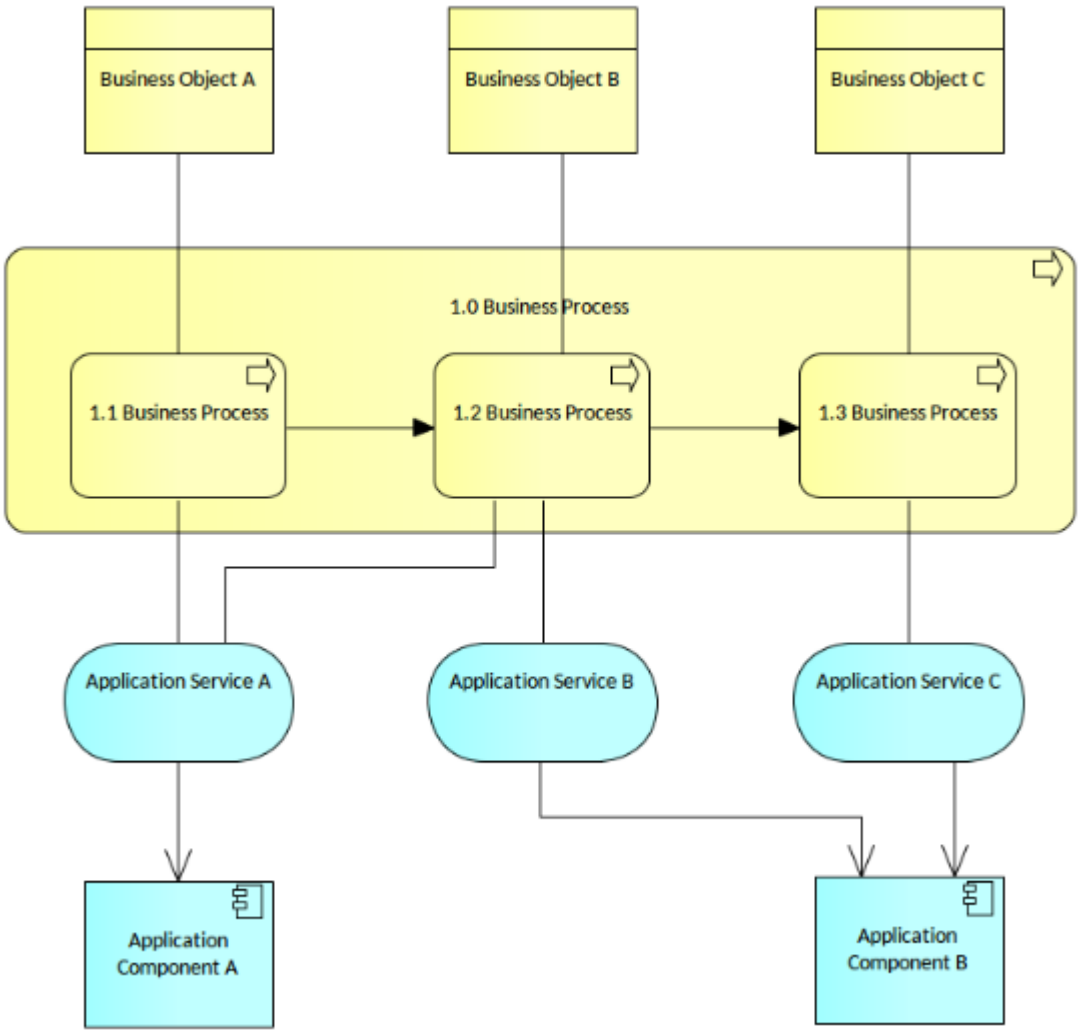


Figure: Showing the *Application Usage* Viewpoint

Technology Usage Viewpoint

The *Technology Usage Viewpoint* pattern creates elements that show how applications are supported by the software and hardware technology: the technology services are delivered by the devices; system software and networks are provided to the applications. This viewpoint plays an important role in the analysis of performance and scalability, since it relates the physical infrastructure to the logical world of applications.

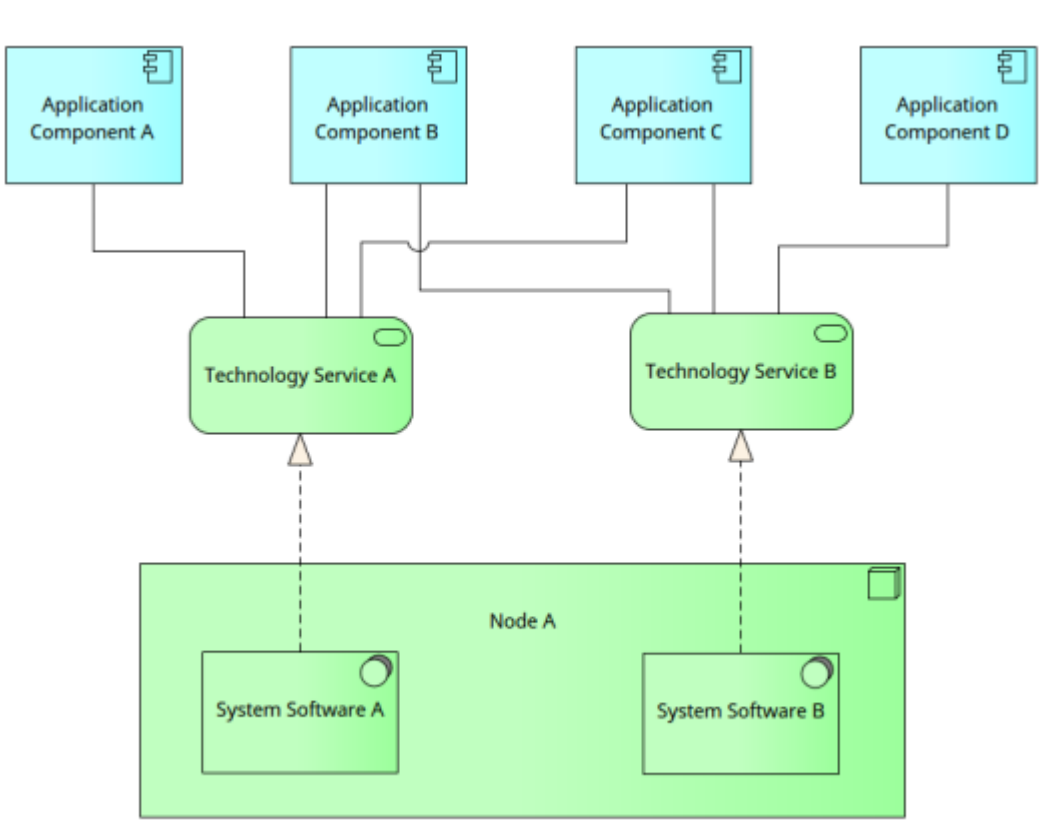


Figure: Showing the *Technology Usage* Viewpoint

Business Process Cooperation Viewpoint

The *Business Process Cooperation Viewpoint* pattern creates elements and a diagram that describe the business processes showing how they relate to each other and also with their environment. This includes relationships with Business Services and Business Objects and the Roles and Actors that perform the processes or who are affected by them.

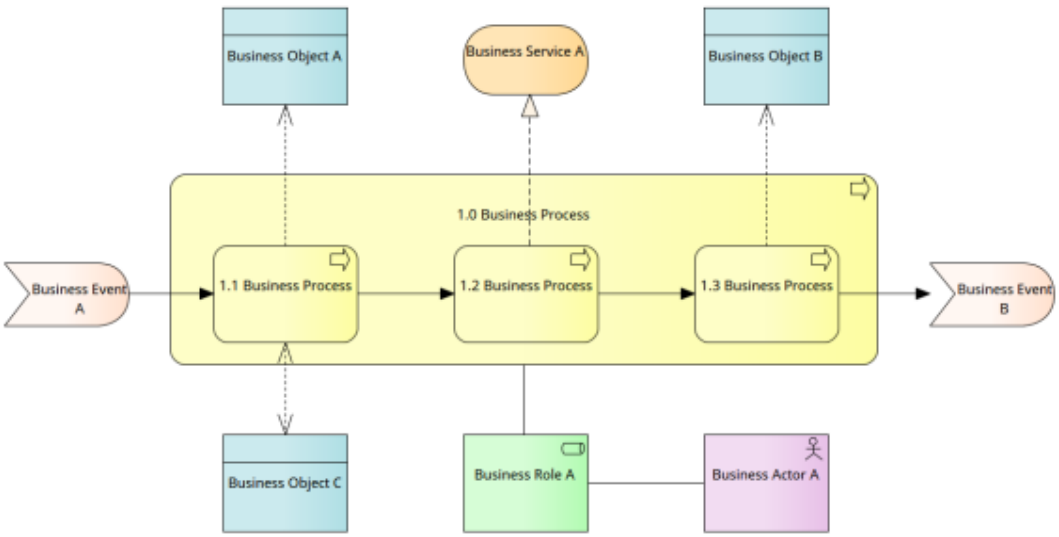


Figure: Showing the Business Process Cooperation Viewpoint

Application Cooperation Viewpoint

The *Application Cooperation Viewpoint* pattern creates elements a diagram that describe the relationships between applications components and their locations, the services they provide or utilize and the information that flows between them.

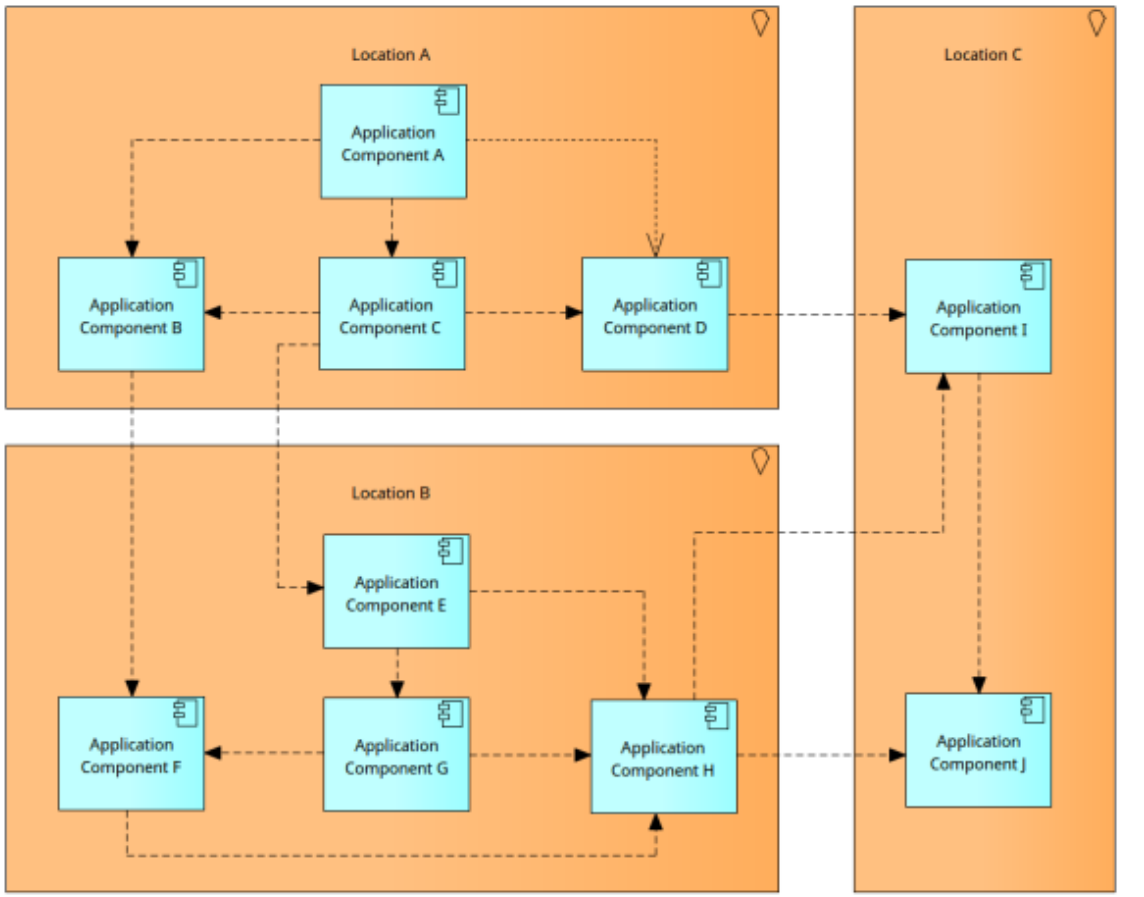


Figure: Showing the *Application Cooperation* Viewpoint

Service Realization Viewpoint

The *Service Realization Viewpoint* pattern creates elements that show how one or more business services are realized by the underlying processes (and sometimes by application components). Thus, it forms the bridge between the business products viewpoint and the business process view. It provides a “view from the outside” on one or more business processes.

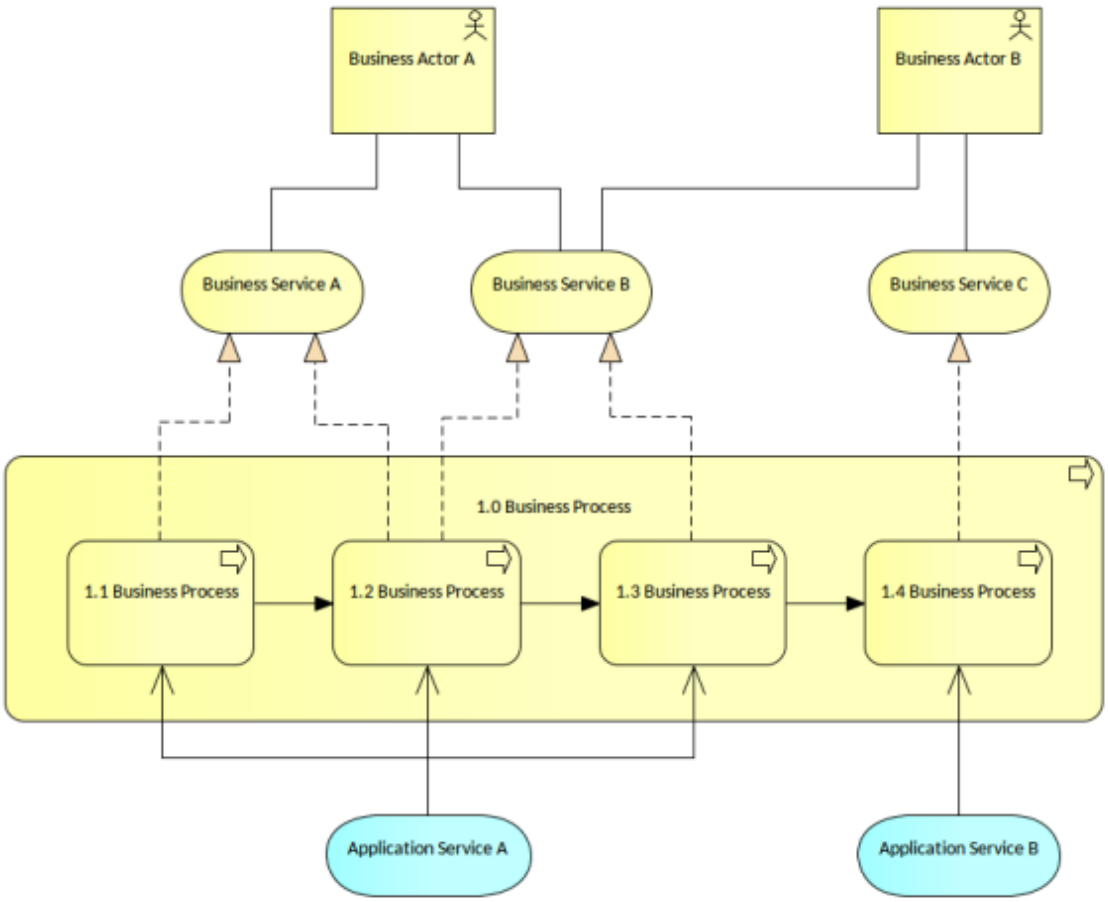


Figure: Showing the *Service Realization* Viewpoint

Implementation and Deployment Viewpoint

The *Implementation and Deployment Viewpoint* pattern creates elements and a diagram that relate programs and projects to the parts of the architecture that they implement. This view allows modeling of the scope of programs, projects and project activities in terms of the plateaus that are realized or the individual architecture elements that are affected. In addition, the way the elements are affected can be indicated by annotating the relationships.

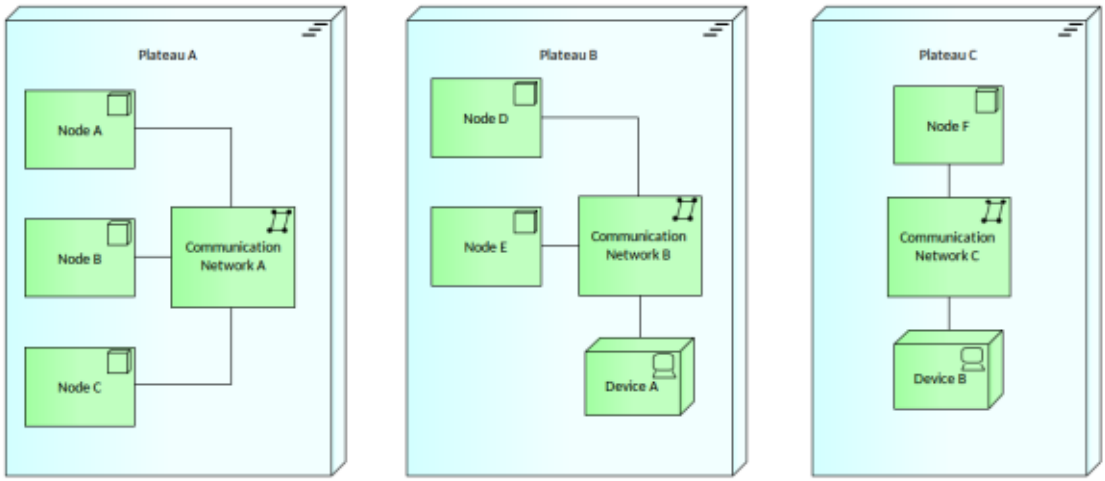


Figure: Showing the *Implementation and Deployment* Viewpoint

Motivation Viewpoints

Table of Motivation Viewpoints

Name	Description
Stakeholder	Focuses on modeling the stakeholders, drivers, the assessments of these drivers, and the initial goals to address these drivers and assessments.
Goal Realization	Focuses on modeling and analyzing the influence relationships between goals (and requirements).
Requirements Realization	Focuses on modeling the realization of requirements and constraints by means of core elements, such as actors, services, processes, application components, etc.
Motivation	Covers the entire motivational aspect and allows use of all motivational elements.

Stakeholder Viewpoint

The *Stakeholder Viewpoint* pattern creates stakeholders, the internal and external drivers for change, and the assessments (in terms of strengths, weaknesses, opportunities, and threats) of these drivers. Also, the links to the initial (high-level) goals that address these concerns and assessments can be described. These goals form the basis for the requirements engineering process, including goal refinement, contribution and conflict analysis, and the derivation of requirements that realize the goals.

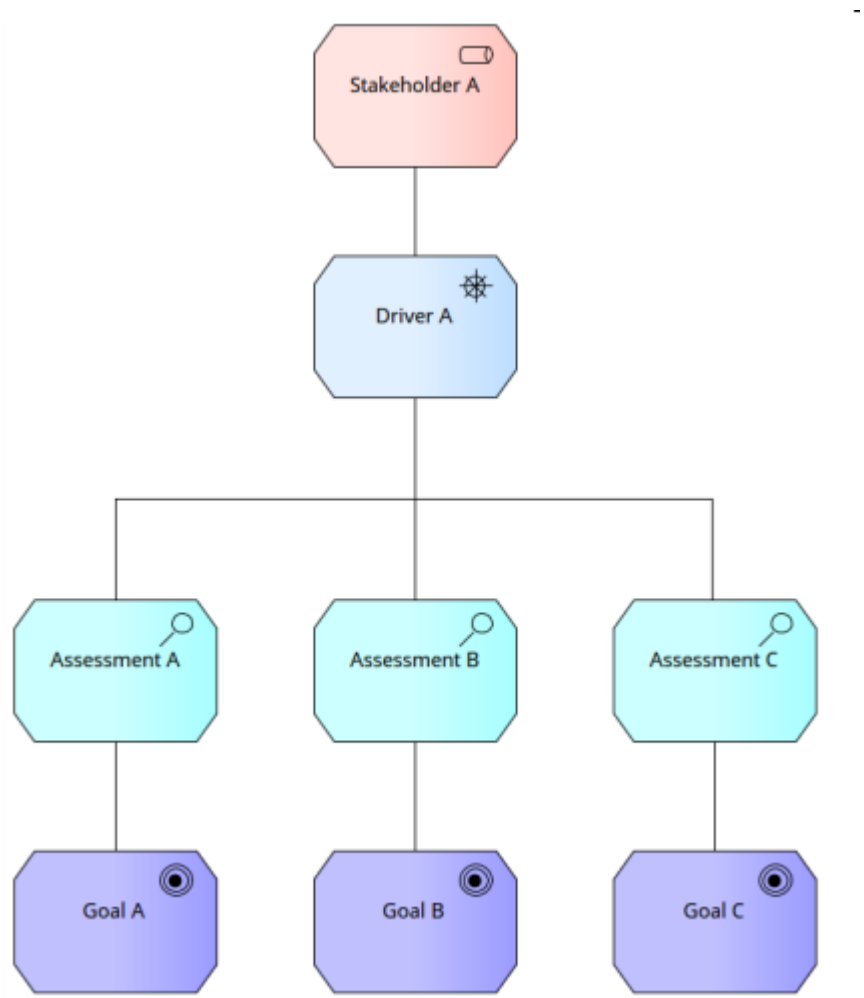


Figure: Showing the *Stakeholder* Viewpoint

Goal Realization Viewpoint

The *Goal Realization Viewpoint* pattern creates elements and a diagram that models the relationships between goals including the decomposition to sub-goals. The goals are realized by an Outcome which is realized by a Principle which behaves like a more abstract and broader requirement. Finally the Principle is realized by a Requirement indicating specific properties that the system must exhibit.

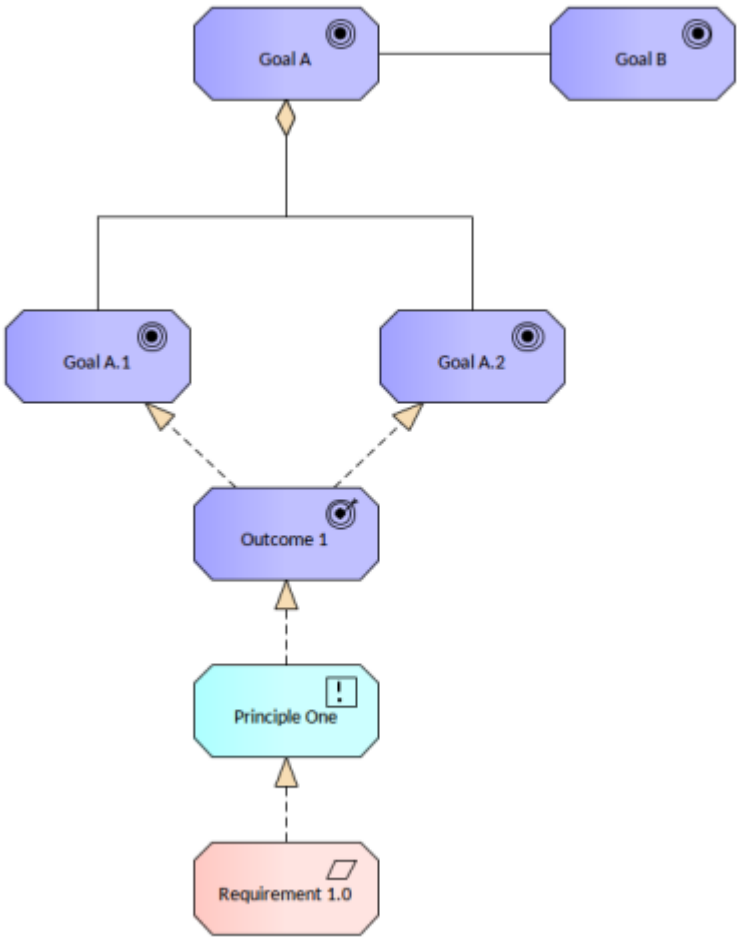


Figure: Showing the *Goal Realization* Viewpoint

Requirements Realization Viewpoint

The *Requirements Realization Viewpoint* pattern creates elements and a diagram that model the realization of Goals into Requirements and Constraints and then how these Requirements are realized by core elements such as Business and Application Services. Color has been introduced to add appeal to the diagram and to distinguish the element types.

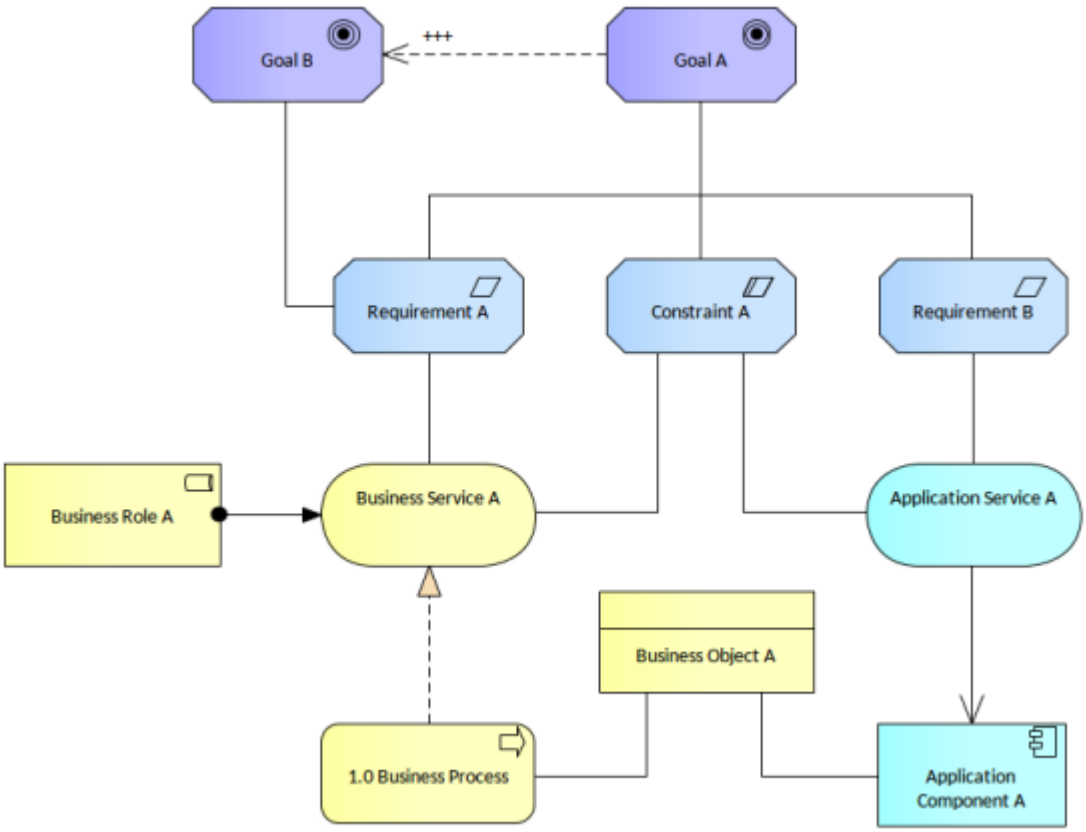


Figure: Showing the *Requirements Realization* Viewpoint

Motivation Viewpoint

The *Motivation Viewpoint* pattern creates elements and a diagram that completely covers the motivational aspect from a given stakeholder's perspective defining a Driver, an Assessment, a number of Goals and the Principle that is applied and the Requirements and Constrains that are needed to qualify the Principle.

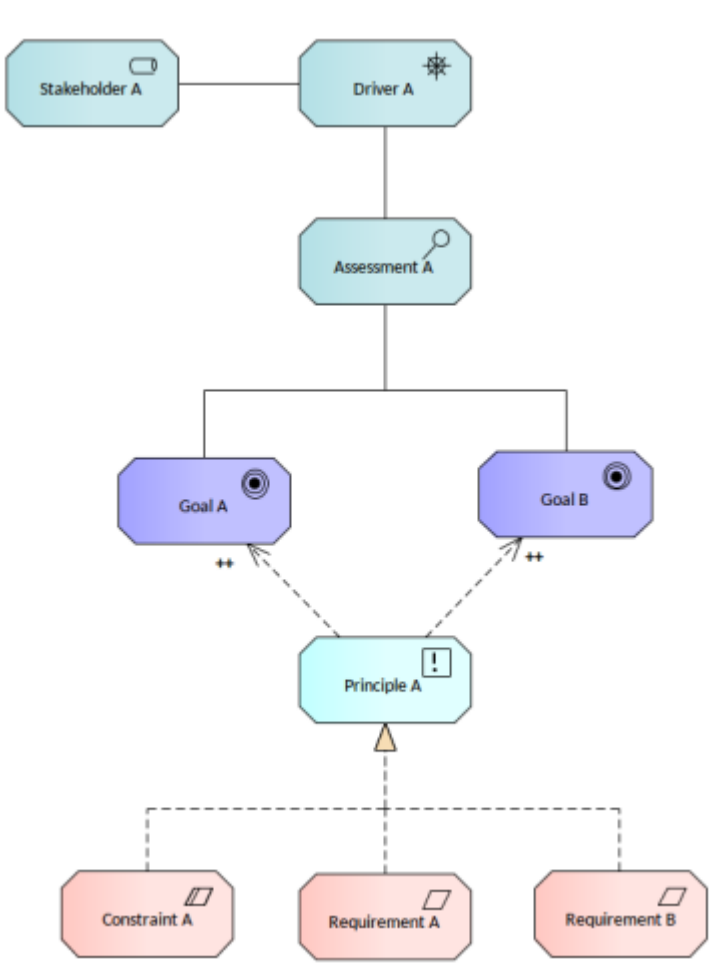


Figure: Showing the *Requirements Realization* Viewpoint

Strategy Viewpoints

Table of Strategy Viewpoints

Strategy	Provides a high-level strategic overview of the strategies of the enterprise, its capabilities, value streams, and resources, and the envisaged outcomes.
Capability Map	Provides an overview of the capabilities of the enterprise.
Value Stream	Shows an overview of value-creating steps in the enterprise and the capabilities that support these.
Outcome Realization	Describes how high-level, business-oriented results are produced by the capabilities and resources of the enterprise.
Resource Map	Provides a structured overview of the resources of the enterprise.

Strategy Viewpoint

The *Strategy Viewpoint* pattern creates elements and a diagram that models the strategic intent of an organization by articulating a Course of Action and the Capabilities and Resources needed to achieve it providing a modeled Outcome.

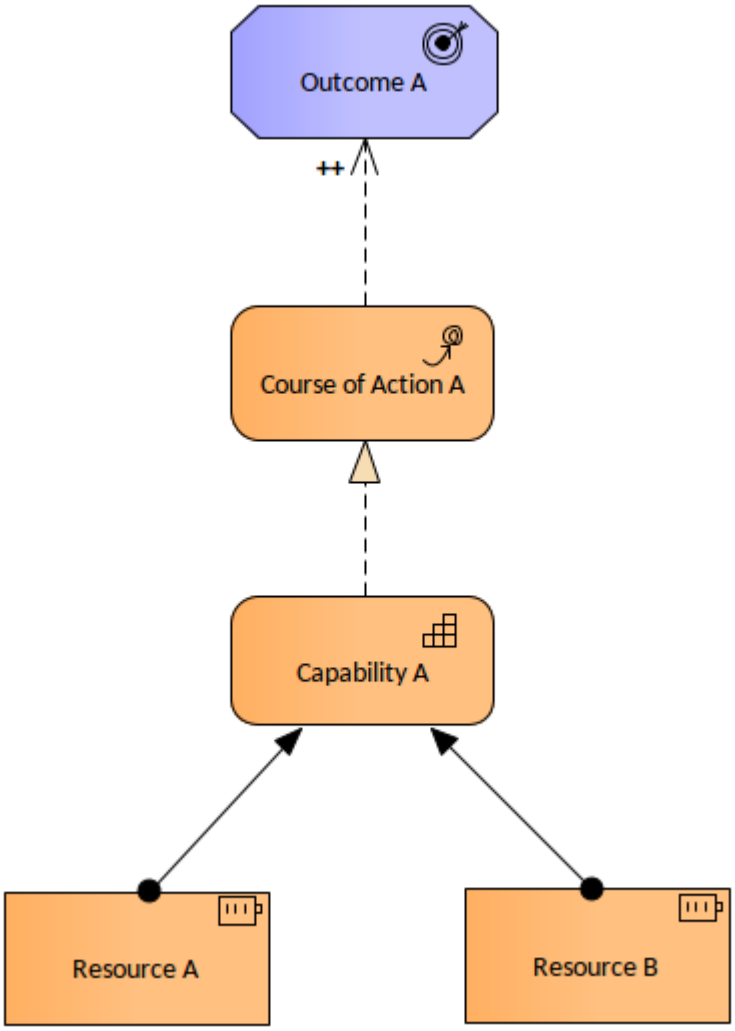


Figure: Showing the *Strategy* Viewpoint

Capability Map Viewpoint

The *Capability Map Viewpoint* pattern creates elements and a diagram that allows Capabilities to be visualized in a nested hierarchy. The Capabilities are also nested in a hierarchy in the Project Browser allowing groups of them to be easily moved from one location to another. Color has been used to convey the levels of the hierarchy.

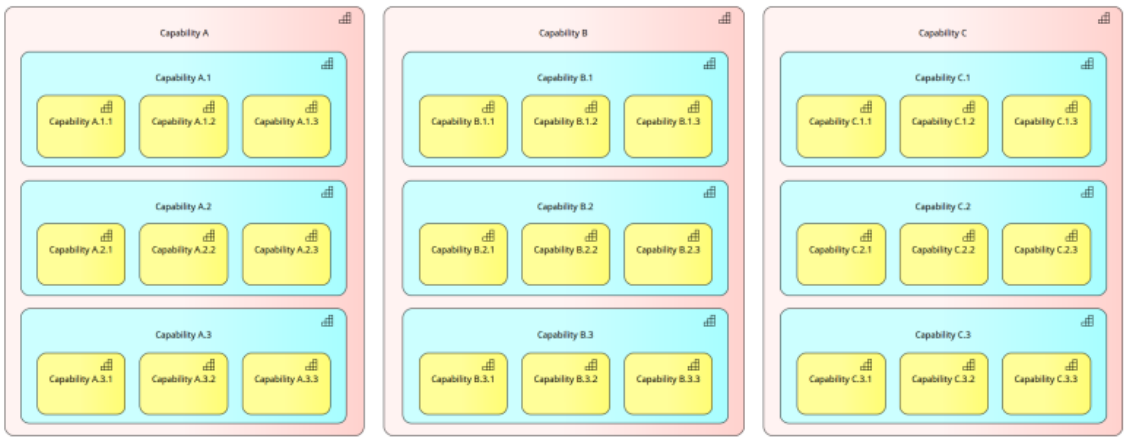


Figure: Showing the *Capability Map* Viewpoint

Value Stream Viewpoint

The value stream viewpoint allows the Business Architect to create a structured overview of a value stream, the capabilities supporting the stages in that value stream, the value created, and the stakeholders involved.

This image shows a diagram created from the Viewpoint Pattern that is a built-in part of the Model Pattern facility.

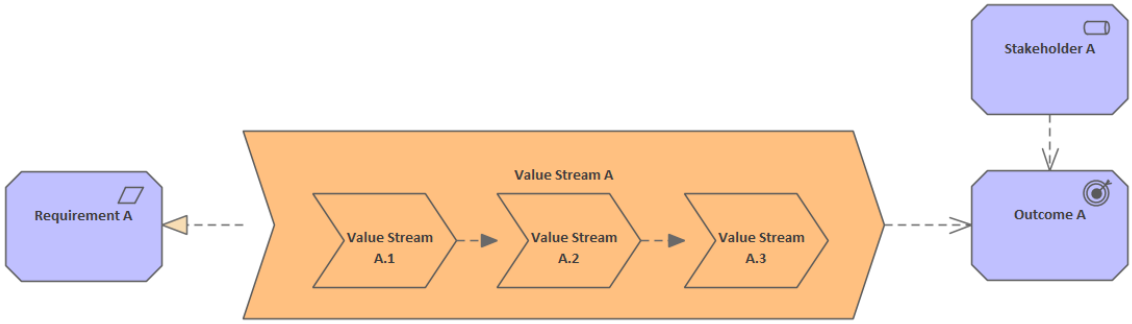


Figure: Showing the *Value Stream* Viewpoint

Outcome Realization Viewpoint

The *Outcome Realization Viewpoint* pattern creates elements and a diagram that model how core elements deliver the high level business value. The diagram is useful to show how the strategic level business elements such as Value and Outcomes are realized by the underlying elements that deliver that value such as Capabilities, Services and Components.

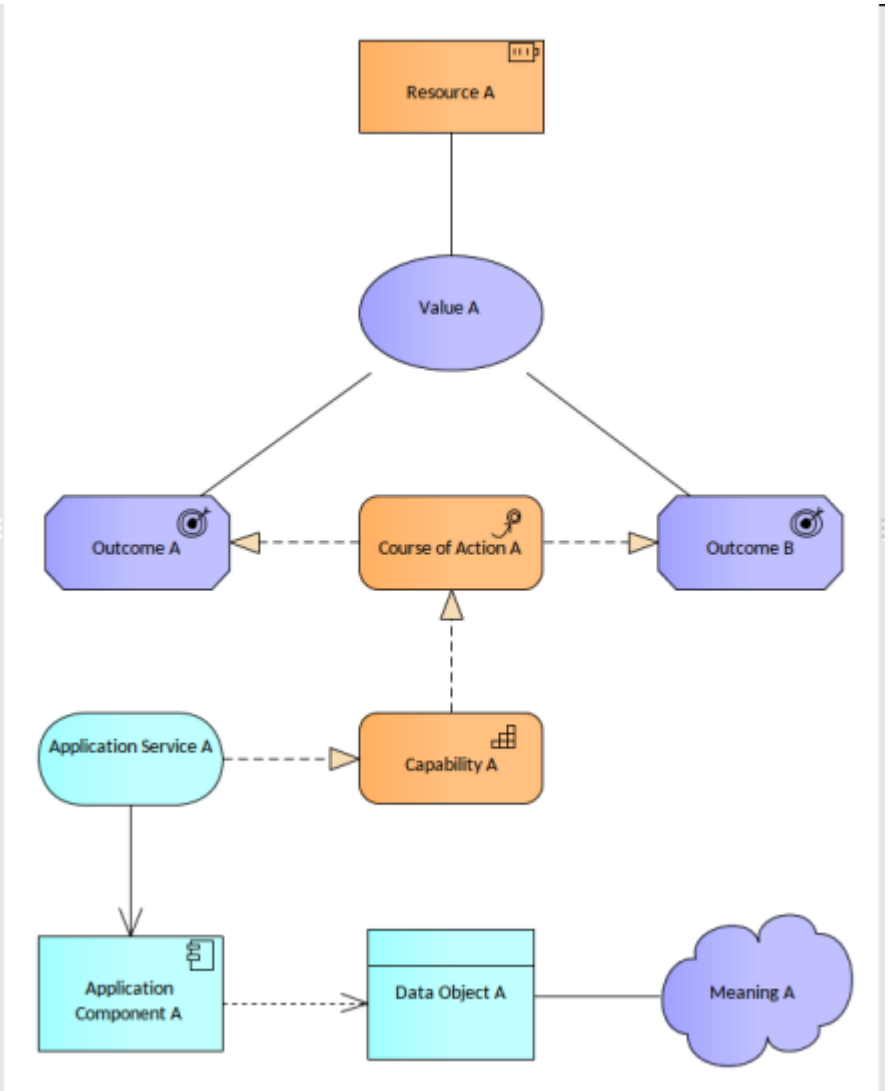


Figure: Showing the *Outcome Realization* Viewpoint

Resource Map Viewpoint

The *Resource Map Viewpoint* pattern creates *a number of Resource elements nested into three layers* . It permits a Business Architect or other stakeholder to create a structured overview of the resources available to an enterprise. The map commonly shows two or three levels of resources across an entire enterprise.

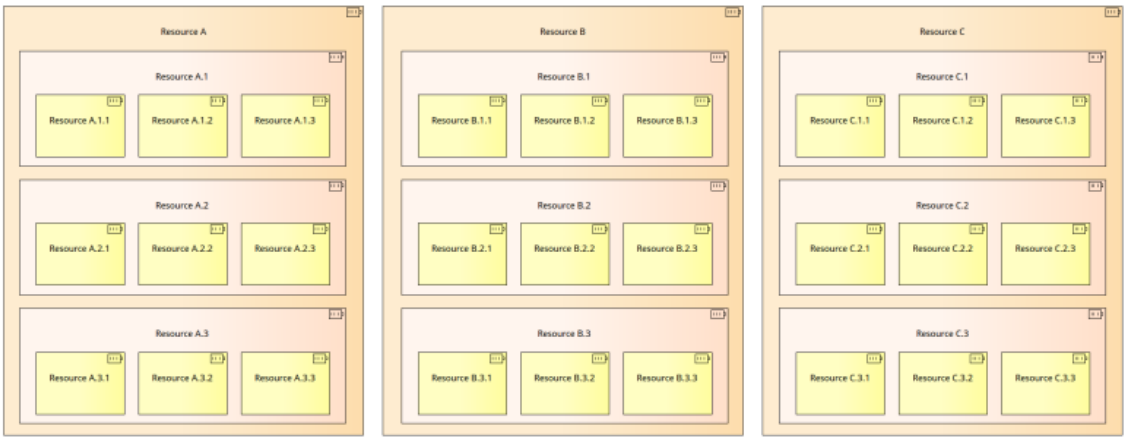


Figure: Showing the *Resource Map* Viewpoint

Implementation and Migration Viewpoints

The Implementation and Migration Viewpoints are designed to allow the modeler to create relevant and meaningful views of the implementation and migration aspects of the architectures including such things as change and transitions.

Table of Implementation and Migration

Project	Primarily used to model the management of architecture change.
Migration	Used to model the transition from an existing architecture to a target architecture.
Implementation and Migration	Used to model the relationships between the programs and projects and the parts of the architecture that they implement.

Project Viewpoint

The *Project Viewpoint* pattern creates elements and diagrams that contains elements that model the management of architecture change. This includes the transition from a baseline to a target enterprise architecture is complex and can be constrained by Portfolio Management, Project Management and a number of other disciplines.

Figure: Showing the *Project* Viewpoint

Migration Viewpoint

The *Migration Viewpoint* pattern creates elements and a diagram that model the transition from a baseline to a target enterprise architecture. The Plateau represents a relatively stable state of the architecture that exists during a limited period of time, whereas the Gap represents a statement of the difference between the two states.

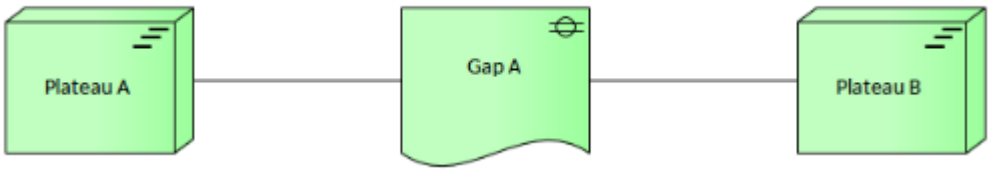


Figure: Showing the *Migration* Viewpoint

Implementation and Migration Viewpoint

The *Implementation and Migration Viewpoint* pattern creates elements and a diagram that model relate programs and projects to the parts of the architecture that they implement. This view allows modeling of the scope of programs, projects, project activities in terms of the plateaus that are realized or the individual architecture elements that are affected.

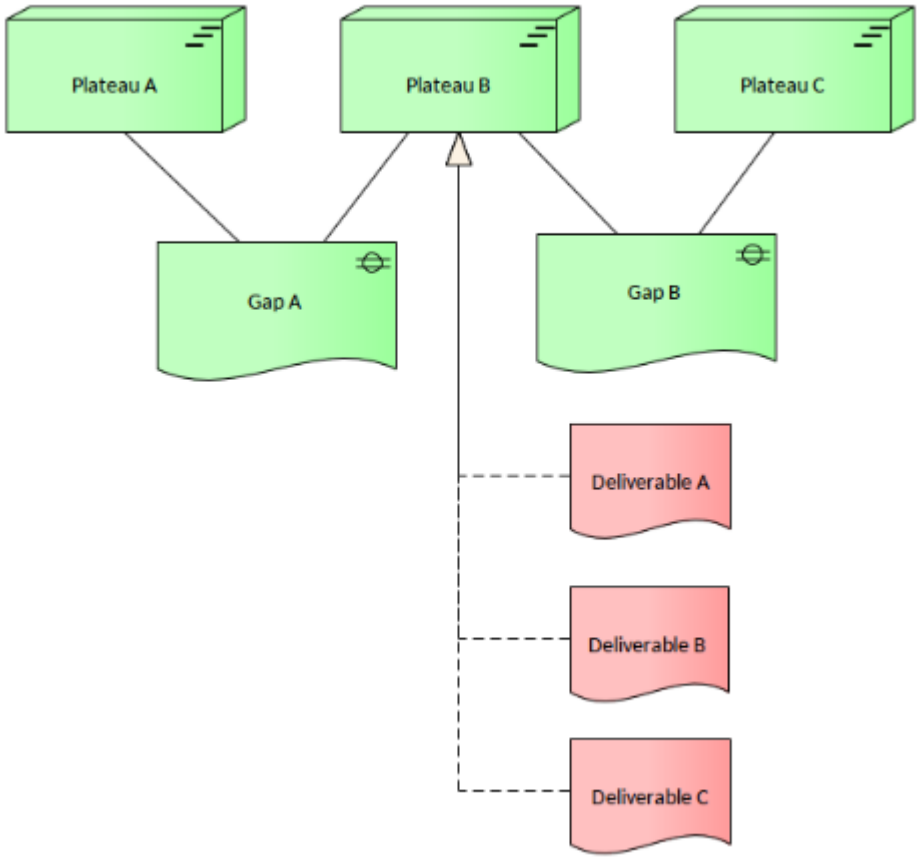


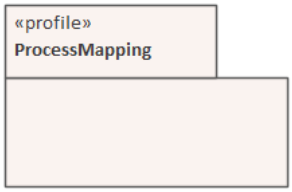
Figure: Showing the *Implementation and Migration* Viewpoint

Custom ArchiMate Viewpoints

This topic discusses the procedures for creating a custom ArchiMate Viewpoint.

Create a Profile with a Viewpoint

All language customization functions within Enterprise Architect start with a Profile Package, as shown in this diagram.



Stage	Description
Select UML Perspective	New Viewpoints are created using the UML Profile mechanism within Enterprise Architect. In order to create a Viewpoint the current Perspective must include the UML Structural elements.
Create a Profile	Drop a Profile from the 'Package' page of the Toolbox. The name given to this item will be visible to users when selecting a Viewpoint to use. When prompted, create a Class diagram for the Package.
Create a View Specification	Open the diagram you have just created. This will also display the 'Profile' page of the Toolbox. Drop a View Specification from the 'Metamodel' section of the Toolbox. The name given to this item will be visible to users when selecting a Viewpoint to use. In practice, you can include multiple View Specifications within your Profile.

Extending a Base ArchiMate Diagram Type

The first part of defining a Viewpoint in Enterprise Architect is defining which diagram type or types it will extend. This selection determines where users will be able to find your Viewpoint and provides the grouping of element and relationship types within the Toolbox.

This diagram demonstrates the definition of a Viewpoint extending an ArchiMate Business Layer diagram.

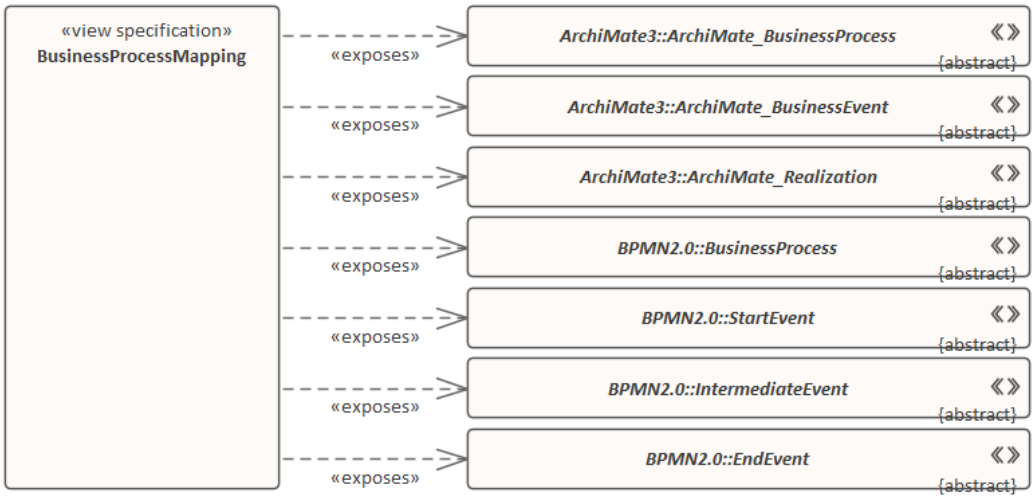
Stage	Description
Create a New Stereotype	Drop a new Stereotype element from the Toolbox. Use the Properties window to mark the Stereotype element as 'Abstract', to indicate that it isn't being defined within this profile.

	To extend one of the diagram types defined for ArchiMate by Enterprise Architect, give the stereotype one of these names: • ArchiMate3::Business • ArchiMate3::Application • ArchiMate3::Technology • ArchiMate3::Motivation • ArchiMate3::Implementation
Create an Extension Connector	Each Viewpoint must be an extension of one or more diagram types. This is specified by creating an Extension connector from the View Specification to the element representing the target diagram type.

Specifying the Elements and Relationships in a Viewpoint

The elements appearing in a Viewpoint are defined by the Exposes relationships from the View Specification to the Stereotypes and UML Metaclasses to be included.

When defining a Viewpoint for a modeling language that you are not the author of, these stereotypes will appear with qualified names, as shown in this diagram.



Stage	Description
Create References to ArchiMate Types	Drag a Metaclass element from the Toolbox onto your diagram. On the resulting dialog, go to the 'Stereotypes' page to select the ArchiMate3 profile. You can then add as many ArchiMate element and relationship types as you require.
Optionally Add References to Non-ArchiMate Types	You can even add types from multiple modeling languages to your Viewpoint. Simply repeat the process of adding a Metaclass element from the Toolbox to select types from any other modeling language supported by Enterprise Architect.
Add Exposes Relationships	Add an Exposes relationship from your View Specification to each of the types that you want displayed in the Viewpoint.

Importing a Profile

Option	Description
Import Package as UML Profile	If you have developed the profile for your Viewpoint in the model that you want to use it in, the easiest way to start using it is to directly import it into the model. To do this, select the profile in the Browser and, in the ribbon, select Specialize > Technologies > Publish Technology > Import Package as UML Profile. This automatically wraps your profile into an MDG Technology and imports it into the model so that all users of the model have it available.
Deploy MDG Technology	If you want to use your Viewpoint across multiple models you have to export the Profile and generate an MDG Technology to include it. This offers much more flexible deployment options and also allows you to include additional customizations, such as allowing you to define model patterns, scripts and reporting templates to complement the Viewpoint definition described here.

Selecting your Viewpoint

Option	Description
For a New Diagram	Within the 'New Diagram' dialog, any diagram type that offers Viewpoints will allow the type to be expanded and a Viewpoint selected.
In Diagram Properties	In the docked Properties window for a diagram, the defined Viewpoints are available in the 'Applied Metamodel' drop-down. In the 'Properties' dialog, the defined Viewpoints are available in the 'View' drop-down.

Using ArchiMate with Enterprise Architecture

With the ArchiMate perspective selected, as shown in the *Getting Started* topic, all of the language features such as concepts, diagrams and views and viewpoints will be available to the modeler. Enterprise Architect also provides a way of creating a repository structure using Packages that will act as the containers for the elements and diagrams that you create to describe your enterprise. There is also a wide range of tools that are useful for working with ArchiMate models including diagram filters, legends, notes and tools for navigating and searching, which will be useful as your models get larger.

Our world has fundamentally shifted to architects working together, often in distributed settings, and this is important as Enterprise Architecture models often grow organically with a number of architects contributing to a central model. Enterprise Architect is essentially a collaboration platform that allows architects and other stakeholders and contributors to work together, sharing ideas and using the discussion, review and other collaboration features to ensure that robust and relevant architectures are created.

Adding Diagrams

Diagrams are one of the most useful ways of communicating with other team members and with stakeholders who have an interest in the enterprise architecture. Diagrams can be created in different ways:

- An empty diagram can be created and existing elements can be added from the Browser window, or new elements and connectors can be added from the Diagram Toolbox
- A diagram can be created from a user-defined pattern that also contains elements and connectors

In the next section we will also explore another method by which a diagram can be created using the Model Wizard tool. As a modeler you are likely to use all of these methods at different times, depending on the circumstances and the modeling context. To create a new ArchiMate diagram you can use one of these methods, ensuring that you have chosen the ArchiMate perspective.

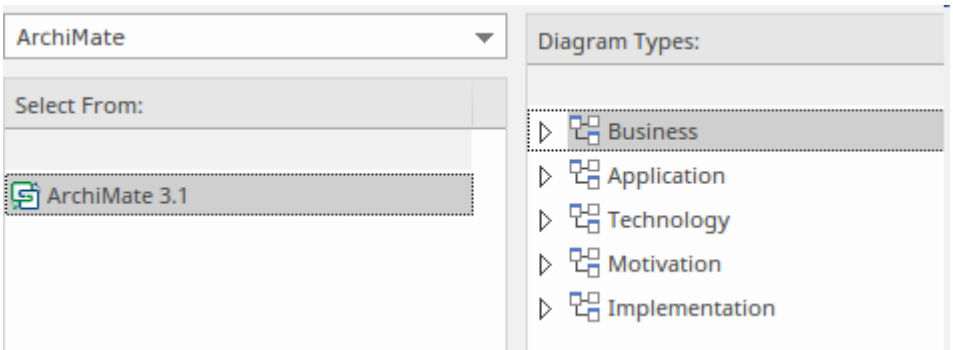


Figure: New Diagram dialog showing the ArchiMate diagram types

Access

Ribbon	Design > Diagram> Add
Context Menu	Browser window | Right-click on Package | Add a Diagram...
Keyboard Shortcuts	<Ctrl> + <Insert>

Creating a Package Structure

There are two approaches to creating a suitable Package structure within Enterprise Architect, both of which mimic the enterprise architecture methods.

- An initial well developed Package structure that is changed very little through the course of model development for an initiative
- A skeleton model that contains the main Packages and is augmented as new needs are understood, and changed significantly during an initiative

Either of these methods can be used, or a team could consider a hybrid approach; either way new Packages need to be created in Enterprise Architect that will act as the containers for new elements and diagrams.

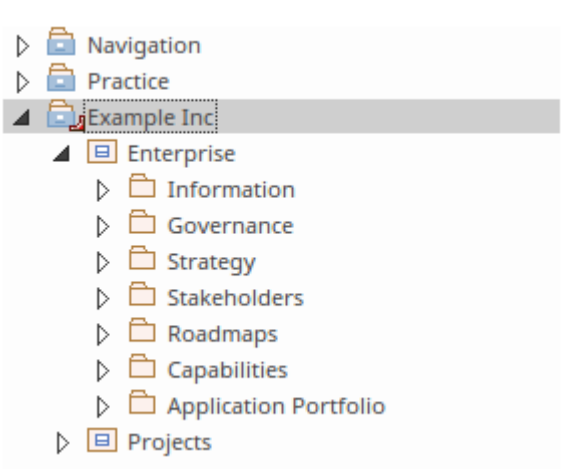


Figure: Showing the Package structure in the 'Project' tab of the Browser window.

Adding Views and Viewpoints

The ArchiMate language defines a series of example viewpoints designed to provide representations that are meaningful and relevant to a variety of stakeholders. These viewpoints are made available in Enterprise Architect through the New Diagram Dialog or the Model Wizard patterns, which provide a way of creating both repository content and diagrams that show how the elements are connected by relationships. Thus new diagrams can be created.

- A new view can be created using the New Diagram Dialog with restriction to the slected Viewpoint.

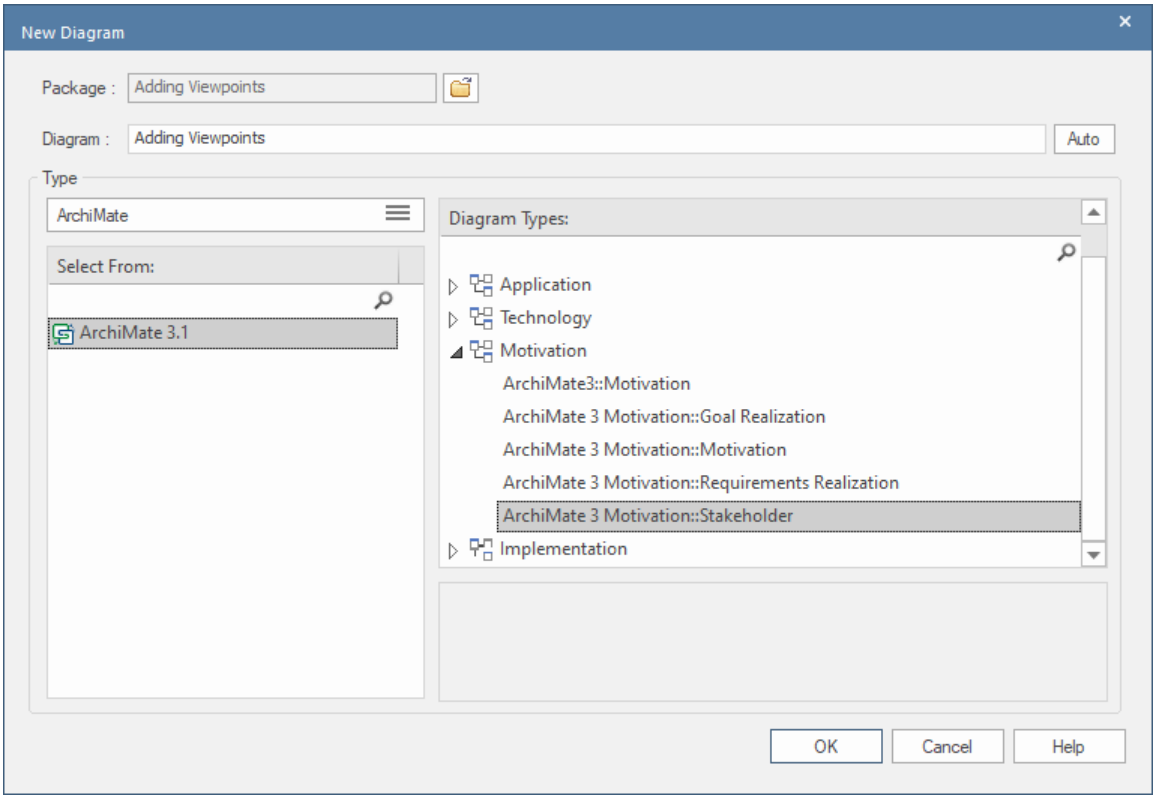


Figure: Showing the New Diagram Dialog for the Stakeholder Viewpoint in the Motivation Viewpoints group.

- A diagram and its elements and connectors can be added using the Model Wizard (Start Page 'Create from Pattern' tab).

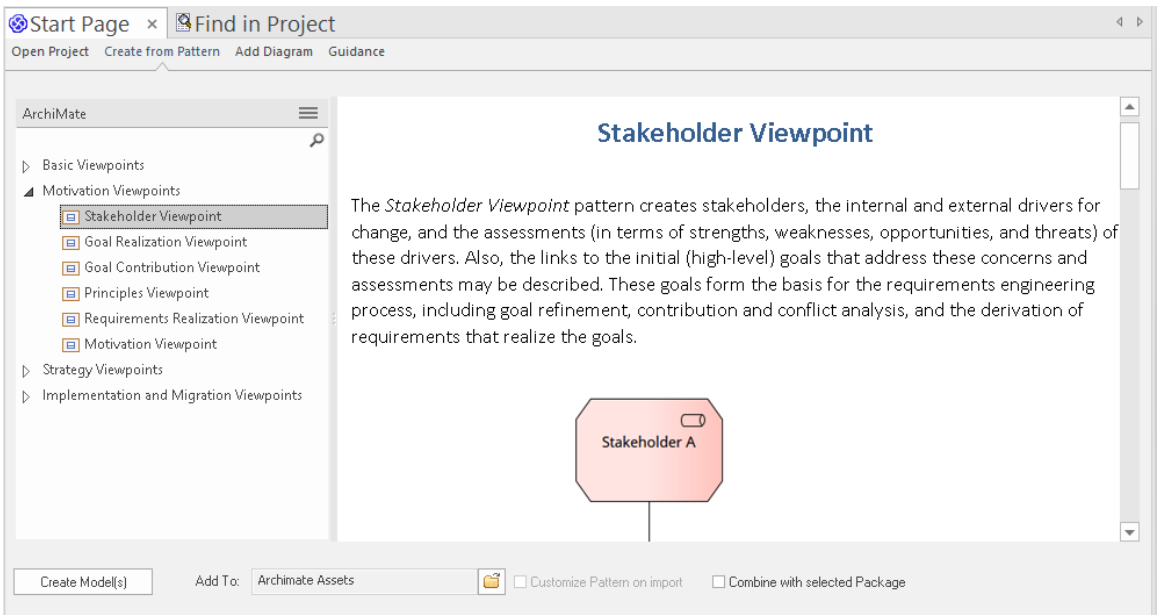
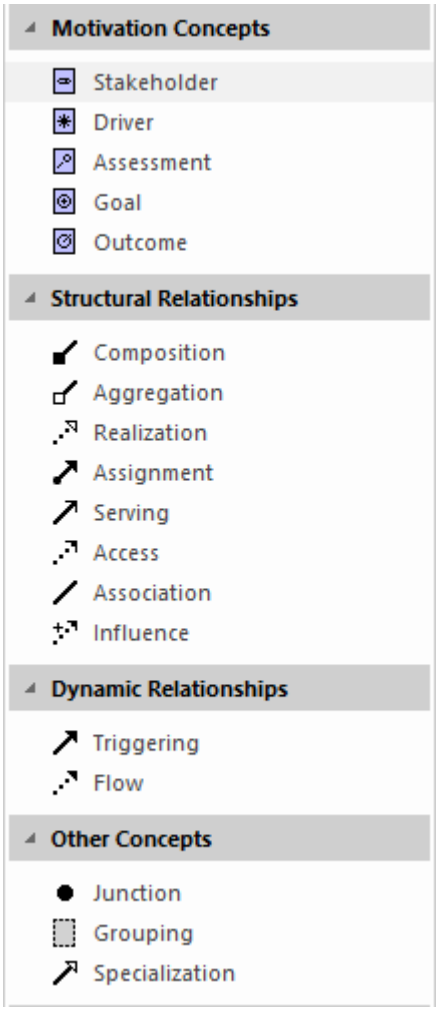


Figure: Showing the Model Wizard and the Stakeholder Viewpoint in the Motivation Viewpoints group.

When the view has been created the diagram toolbox will restrict the elements and connector to only those that are part of the viewpoint.



Adding Elements and Relationships

Elements can be added to the model directly without the need for a diagram to be created, but it is far more common for a diagram to be the device that is used to add both elements and connectors to the model. Diagrams can be built up with a combination of:

- Existing elements dragged from the Browser
- New elements (or Relationships) dragged from the Diagram Toolbox pages

Adding Elements from the Browser

This diagram shows how elements can be added from the Browser window by dragging and dropping them onto the current open diagram canvas.

Figure: Showing an existing element being dragged from the Browser window

Adding Elements from the Toolbox

This diagram shows how elements can be added from the Toolbox pages by dragging and dropping elements (or relationships) onto the current open diagram canvas.

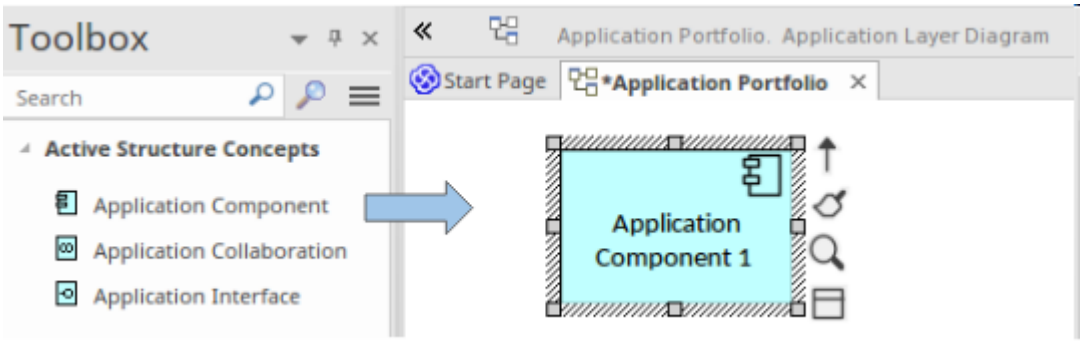


Figure: Showing an existing element being dragged from a Toolbox page

Changing Elements and Relationships

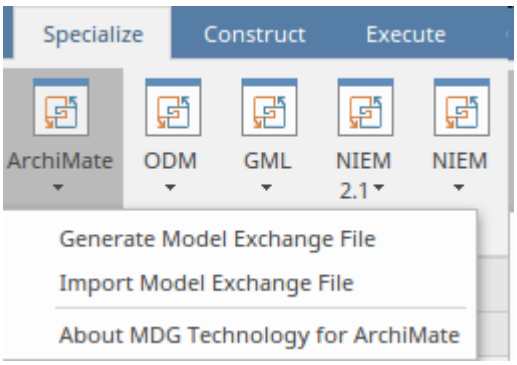
Any element or relationship can be changed including its name and properties. When a change is made to an element in any location, for example in a diagram this change will be reflected in any other diagrams (view) that contains the element or relationship.

Exchanging ArchiMate Models

Enterprise Architect supports model exchange in accordance with the specification and allows architects and others to export and import models that comply with the exchange file format. Enterprise Architect is a multi-featured tool and its price point, flexible licensing model and browser-based access (to the Pro Cloud Server) allows the tool to be made available to a large number of users from interdisciplinary teams.

The tool provides these two exchange options:

- Generate Model Exchange File -*allowing models to be exported from Enterprise Architect*
- Import Model Exchange File - *allowing models to be imported into Enterprise Architect*



Generate Model Exchange File

After you have created an ArchiMate® 2 or ArchiMate® 3.1 model using ArchiMate 2 or ArchiMate 3.1 respectively in Enterprise Architect, you can generate a Model Exchange File from the model Package.

Access

Ribbon	Specialize > Technologies > ArchiMate > Generate Model Exchange File
Context Menu	Right-click on Package | Specialize | ArchiMate | Generate Model Exchange File

Generate Model Exchange File

Option	Action
Package	Displays the name of the currently-selected Package. If this is not the required Package, click on the button and select the correct Package.
Filename	Type in the file path and name of the file to be generated, or click on the button and browse for the location.
Version	Click on the drop-down arrow and select the version of the Model Exchange File.
Language	(Optional) Select the language identifier for the content of the elements in the generated file.
Include	Select the appropriate checkboxes: • 'Element Relationships' - to include connectors between the elements being generated • 'Tagged Values' - to include the Tagged Values for each element and relationship being generated • 'Element Organization' - to include the structural organization of the elements inside the selected Package • 'Diagrams' - include the ArchiMate diagrams inside the selected Package
Generate	Click on this button to generate the Model Exchange File. The progress of the file generation is reported in the 'ArchiMate' tab of the System Output window. A message box also displays to indicate when the generation is complete; click on the OK button to clear the message.
View File	Click on this button to display the contents of the generated file.
Close	Click on this button to close this 'Generate Model Exchange File Format' dialog.

Help	Click on this button to display this Help topic.

Notes

- This facility is available in the Professional, Corporate, Unified and Ultimate Editions of Enterprise Architect
- The 'Language' field displays the 2-letter codes from the Language Subtag Registry: http://www.iana.org/assignments/language-subtag-registry/language-subtag-registry
- The Model Exchange File generated by Enterprise Architect adheres to The Open Group Specified Schema
- Generated elements and diagrams will be displayed in the System Output window – double-click on an item in the System Output window to highlight it in the Browser window
- Elements, connectors and diagrams that are not from ArchiMate 2 or ArchiMate 3.1 will be ignored during generation
- The Network and Communication Path connectors from the 'ArchiMate 2::Technology' toolbox in Enterprise Architect will not currently be generated as they do not have any mapping in The Open Group Specified Schema
- The Technology <object> elements, Path connector, Communication Network connector and Distribution Network connector from the 'ArchiMate 3.1 Technology' toolbox in Enterprise Architect will not currently be generated as they do not have any mapping in The Open Group Specified Schema

Import ArchiMate Model Exchange File

If you have a valid ArchiMate® 2 or ArchiMate® 3.1 Model Exchange File, you can import it into your Enterprise Architect project as a UML model.

Access

Ribbon	Specialize > Technologies > ArchiMate > Import Model Exchange File
Context Menu	Right-click on Package | Specialize | ArchiMate | Import Model Exchange File

Import Model Exchange File

Option	Action
Package	Displays the name of the currently-selected Package. If this is not the required Package, click on the button to select the Package Browser and select the correct Package.
Filename	Type in the file path of the file to be imported, or click on the button and browse for the location.
Connector Style	Select one of these styles to set on the imported connectors: • Orthogonal – Rounded (rounded corners) • Orthogonal – Square (square corners) • Custom
View File	Click on this button to open and view the file to be imported.
Import	Click on this button to import the selected Model Exchange file into the specified Package. The progress of the file import is reported in the 'ArchiMate' tab of the System Output window. A message box also displays to indicate when the import is complete; click on the OK button to clear the message.
Close	Click on this button to close the dialog.
Help	Click on this button to display this Help topic.

Notes

- This facility is available in the Professional, Corporate, Unified and Ultimate Editions of Enterprise Architect

- The Model Exchange File should adhere to The Open Group Specified Schema
- Elements, connectors and diagrams that are not specified in *The Open Group Schema* will be ignored during import
- Imported elements and diagrams will be displayed in the System Output window – double-click on an item in this window to highlight it in the Browser window

Migrate to the Latest ArchiMate Version

If you have created a model under one version of ArchiMate, you can migrate all or part of it to a later version; for example, migrate the model from ArchiMate 2 to ArchiMate 3.1. You perform the migration using the Automation Project Interface function Migrate(), which updates the Tagged Values and, if required, stereotypes to the later version of ArchiMate for all elements, attributes, connectors and diagrams under the selected Package or element.

When you migrate your model, you must have both releases of the Technology enabled; when the migration is complete, disable the older release of the Technology.

Migrate ArchiMate 2 model to ArchiMate 3.1

In Enterprise Architect, ArchiMate 3.1 has a migration script built into it, to upgrade a model built for ArchiMate 2 to ArchiMate 3.1. To execute this script:

1. Select the 'Specialize > Tools > Scripting' ribbon option.
2. Expand the *ArchiMate® 3.1* folder.
3. Right-click on the 'Migrate ArchiMate 2 to ArchiMate 3' script.
4. Click on the 'Run Script' option from the context menu.
5. Monitor the execution of the script in the System Output window.

When the script has finished executing, disable the ArchiMate 2 technology.

Notes (Migrate ArchiMate 2 to ArchiMate 3.1)

- If you want to migrate from an ArchiMate model to an ArchiMate 3.1 model, you must do so in two stages: from ArchiMate to ArchiMate 2, and from ArchiMate 2 to ArchiMate 3.1

Migrate ArchiMate model to ArchiMate 2

You must create a script to call the Migrate() function to migrate a Package or element to ArchiMate 2: this is a VB script example of such a script:

```
Sub MigrateElement (sGUID, lngPackageID)
    Dim proj as EA.Project
    set proj = Repository.GetProjectInterface
    proj.Migrate sGUID, "ArchiMate", "ArchiMate 2"
    'refresh the model
    If lngPackageID<>0 Then
        Repository.RefreshModelView (lngPackageID)
    End If
End Sub
Sub MigrateSelectedItem
    Dim selType
    Dim selElement as EA.Element
```

```
    Dim selPackage as EA.Package
    selType = GetTreeSelectedItemType
    If selType = 4 Then 'means Element
        set selElement = GetTreeSelectedObject
        MigrateElement selElement.ElementGUID, selElement.PackageID
        MsgBox "Element Migration Completed",0,"ArchiMate 2 Migration"
    ElseIf selType = 5 Then 'means Package
        set selPackage = GetTreeSelectedObject
        MigrateElement selPackage.PackageGUID, selPackage.PackageID
        MsgBox "Package Migration Completed",0,"ArchiMate 2 Migration"
    Else
        MsgBox "Select a Package or Element in the Browser window to initiate migration",0,"ArchiMate 2 Migration"
    End If
End Sub
Sub Main
    MigrateSelectedItem
End Sub
Main
```

Notes (Migrate ArchiMate to ArchiMate 2)

- All diagrams are converted to Business Layer diagrams
- Collaboration elements are converted to Business Collaboration or Application Collaboration depending on the value of the 'collaborationtype' Tagged Value
- Object elements are converted to Business Object, Contract or Data Object depending on the value of the 'objecttype' Tagged Value
- Interface elements are converted to Business Interface, Application Interface or Infrastructure Interface depending on the value of the 'interfacetype' Tagged Value
- Function elements are converted to Business Function or Application Function depending on the value of the 'functiontype' Tagged Value
- Interaction elements are converted to Business Interaction or Application Interaction depending on the value of the 'interactiontype' Tagged Value
- Service elements are converted to Business Service, Application Service or Infrastructure Service depending on the value of the 'servicetype' Tagged Value
- The 'iconstyle' Tagged Value is removed and the 'Use Rectangle Notation' menu option is set on or off as appropriate
- Process elements become Business Process elements
- Event elements become Business Event elements
- Actor elements become Business Actor elements
- Role elements become Business Role elements
- Component elements become Application Component elements
- Software elements become System Software elements
- Specialisation connectors become Specialization connectors
- Realisation connectors become Realization connectors

- Network Connector connectors become Network
- Node elements now extend UML Class elements
- Software elements now extend UML Class elements
- Junction elements now extend UML Decision elements