SPARX
SYSTEMS

Enterprise Architect

Business Models

Creating Business Models? Sparx Systems Enterprise Architect helps build process models within an overall architecture, providing charts, BPMN Patterns, Technologies, template-driven reports, strategic models, mind maps, and requirements management.

Author: Sparx Systems
Date: 2020-01-20
Version: 15.1

CREATED WITH **ENTERPRISE ARCHITECT**

Table of Contents

Business Models

Change is a constant in business - change in market opportunities, change in source materials, change in customer audience, change in environment - all leading to change in business processes. Re-engineering business processes is a strategic part of any business. The basic requirement that enterprise-class business process engineering principles are hard wired into operations is applied across the spectrum of different business sectors. In fulfilling this fundamental governance requirement, Enterprise Architect and a number of standards, including the Business Process Model and Notation (BPMN), are essential tools.

Enterprise Architect includes a wide range of modeling tools, diagram types, Patterns, Technologies and capabilities to support modeling a wide range of business-focused and analytic domains. From strategic models that support capturing information and goals pertaining to the overall functioning of a large enterprise to simple Mind Maps, process modeling, requirements management, BPMN models and more, Enterprise Architect is the ideal platform for building and sharing business based models that can tightly integrate into an overall enterprise or system level architecture.

In addition to the extensive range of template driven reports available, Enterprise Architect also includes comprehensive support for building management level charts and

dashboards that integrate tightly with model content. Series charts, pie graphs, heat maps and others make it possible to build a high level summary of the state of the current model (and of other models using custom connections).

Business Modeling Overview

This table identifies some of the material to be discussed in the Business Engineering section. Note that many of the capabilities listed here are equally applicable to Systems-, Software- and Enterprise-level modeling and design.

Please note that some features are edition dependent - please consult the Sparx Systems web site for details about which edition supports which features.

Feature	Description
Strategic Models	A range of strategy based model types including Strategy Maps, Value Chain, Balanced Scorecard, Dataflow and Org Chart. Essential modeling for management and for expressing overall business or enterprise level goals and strategy.
Charts and Dashboards	Material supporting the building of dashboards and specification of charts based on model content. Charts provide a real-time view into the health and status

	of a model and allow for hot spots and problem areas to be detected early and dealt with accordingly. A number of chart types are available, including Pie, Series, Bar and Heat Map.
Analysis Models	This section identifies a number of analytic diagram types that have an affinity for business engineering and analysis. Mind Mapping, Process diagrams and custom stereotypes targeting business scenarios and terms are some of the features discussed.
Requirements	Coverage of managing requirements in Enterprise Architect for the business user. A shortened version of the book length "Requirements Engineering" topic also available from Sparx Systems. Covers creating, tracing, managing and generally dealing with all kinds of requirements.
Decision Models	Information about building Decision tables as per the OMG decision modeling notation. Shows how to build a Decision table and how to display that table on a diagram.
Business	An advanced mechanism for creating and

Rules	working with Business Rules in a manner similar to the Decision Tables specified earlier but with some additional functionality that allows for the translation of those rules into activity charts and into code.
BPMN Support	Enterprise Architect has very detailed and comprehensive support for the Business Process Model and Notation specification. BPMN is a popular business and process modeling notation that targets unambiguously defining business (and other) processes in a visual manner with downstream capabilities for simulation, model exchange and code generation.
BPSim Support	BPSim provides the business modeler with a mechanism for defining additional characteristics and parameters for a BPMN model that allow the BPMN model to be simulated with different 'real world' scenarios. Performance data such as time, efficiency, throughput, resources used, resources idle and so on, can be captured and reported from a suitable BPSim engine. Sparx Systems provide a BPSim-capable

	simulator (for separate purchase) - the MDG BPSim Execution Engine - which integrates with the BPSim and BPMN models defined in Enterprise Architect, providing the capability to run and store the results from multiple simulations and to perform convenient comparisons across each configuration's result set. Installing the Sparx Systems MDG BPSim Execution Engine and registering the licence for it are pre-requisites for accessing and using the built-in BPSim configuration facilities.
BPEL	Support for modeling and working with the BPEL (Business Process Execution Language), a subset of BPMN that supports export in the BPEL execution language format.
Business Modeling Notation	Stereotypes and modeling conventions from the UML business modeling extensions defined in earlier versions of the UML.
Eriksson-Penker Notation	Support for the Business Process Model and Notation, defined by Eriksson and Penker. A useful and straightforward process mapping that quickly builds maps

	of processes, inputs, outputs and goals that constitute a business or enterprise level process.
Case Management Model & Notation (CMMN)	The MDG Technology for the Case Management Model & Notation helps you to create CMMN models with Enterprise Architect, for representing actions taken regarding a subject in a particular situation to achieve a desired outcome.

Strategic Models

Enterprise Architect contains a number of diagram types that are specifically aligned with creating high quality strategic business models. Strategic models are typically defined by executives, modelers and managers responsible for defining strategic direction and both short and long term goals.

Advantages of Strategic Models

Defining and visualizing strategy as a series of diagrams/models within an Enterprise Architect repository has numerous advantages over defining that strategy in some other format. First and foremost, by modeling strategy within a repository, you immediately link the strategic direction and executive thinking to the processes, architectures, software and capabilities that are modeled within that same repository. This is a highly effective and powerful means of ensuring that strategic direction is correctly implemented and tied to the fundamental infrastructure and direction of an enterprise.

With strategic direction, goals, decisions and other critical business information defined up front and always accessible to designers, modelers, managers and others working on implementing a strategy, it is significantly more likely that the preferred strategic direction will be adhered to and less likely that simple errors and misunderstandings will occur.

As the solution implementation takes shape, with full trace-ability available as models are constructed, it is possible to trace physical and logical solutions right back through requirements and Use Cases to the actual strategic goals that require that solution to exist. Aligning business needs with infrastructure and capabilities is now much easier and less error prone.

This section will discuss the principal strategic modeling types. Note that there are also significant other model types that can be used to convincingly portray strategy, for example Mind Mapping.
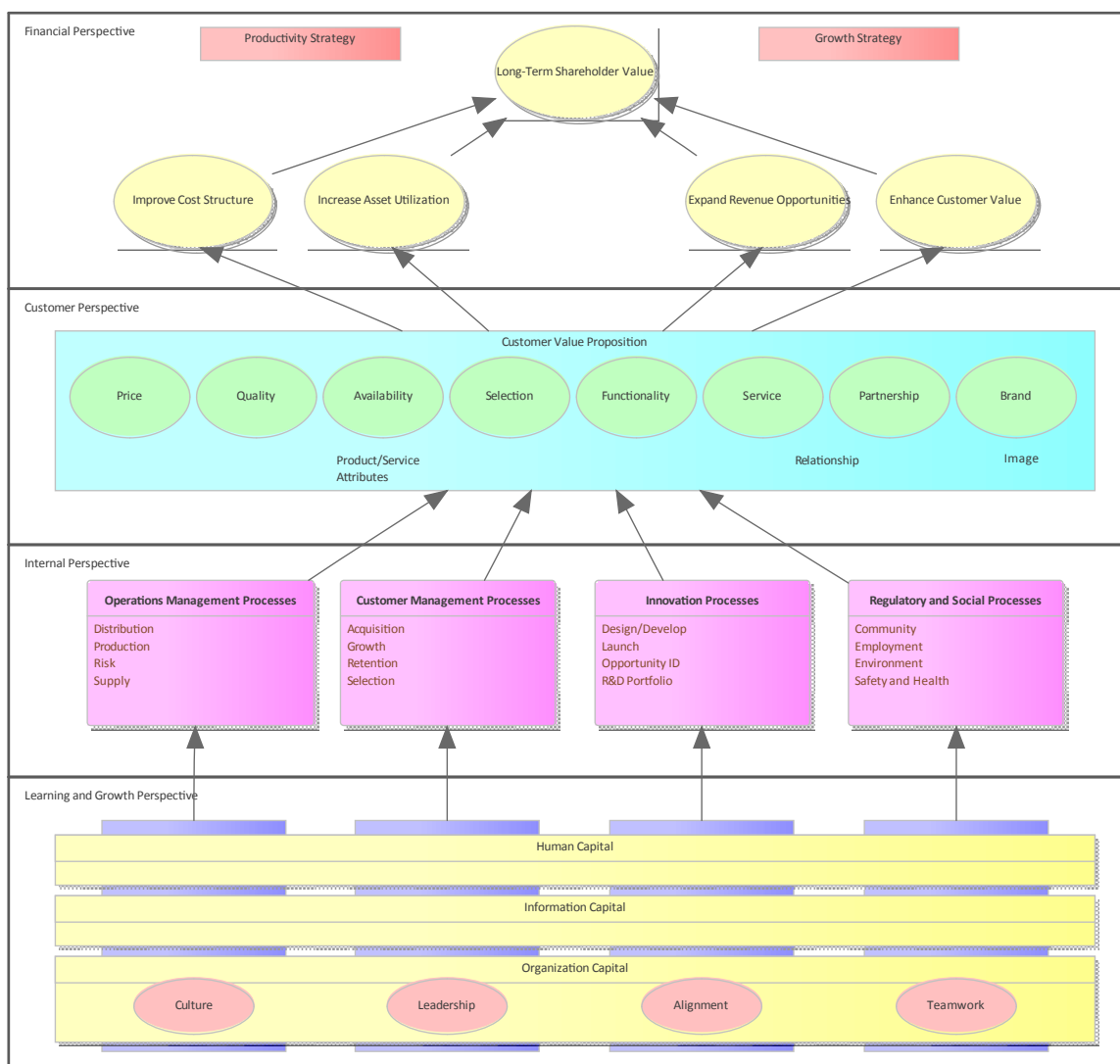
Strategy Map

Kaplan and Norton developed the Balanced Scorecard and the Strategy Map *(Kaplan and Norton, 2004)*. The Strategy Map extends the Balanced Scorecard approach in order to clarify corporate strategy, using four perspectives and their 'cause and effect' relationship. The four perspectives are:

· Financial

· Customer

· Internal Business Process

· Learning and Growth

Strategy Maps are typically used for planning and communicating the overall strategy of a given organization in order to create sustained value for customers and shareholders in the form of a 'cause and effect' chain. Strategy Maps provide a greater emphasis on measuring and managing the value created by enhancing an organization's intangible assets.

Intangible assets are items of value that can not necessarily be seen or touched. Examples of intangible assets include your brand, intellectual property, reputation, good will, patents, creativity and staff morale. A large percentage of an organization's shareholder and customer value is associated with intangible assets, yet traditional financial management systems fail to track, measure or evaluate them. As a result it is important that tools such as a Balanced Scorecard and Strategy Map are used to develop a more balanced approach to Strategic Planning. The ability to see 'cause and effect'

relationships helps an organization understand the chain of events that sees intangible assets transformed into real value.

Strategy Map Patterns

Enterprise Architect provides three basic Patterns that can be used to create a Strategy Map. The first Pattern is very basic and provides the greatest flexibility with respect to the layout of your Strategy Map. The Toolbox provides a number of tools such as a Mission and an Intangible Asset. The second and third Patterns provide a more comprehensive overview, with far more internal processing and Intangible Assets.

Strategic objectives are represented as an oval.

- Create a new Package called 'Strategy Map'

- Create a new Strategy Map diagram called 'Our Strategy'

- Select the Pattern entitled 'Strategy Map 2' and place it onto the Diagram View

- This diagram will display all four perspectives and include a detailed breakdown of each

- Examine the two additional Patterns in order to see the layout of each alternative Strategy Map example

Example Strategy Map (Style 1)

Financial Perspective

Customer Perspective

Internal Perspective

Learning and Growth Perspective
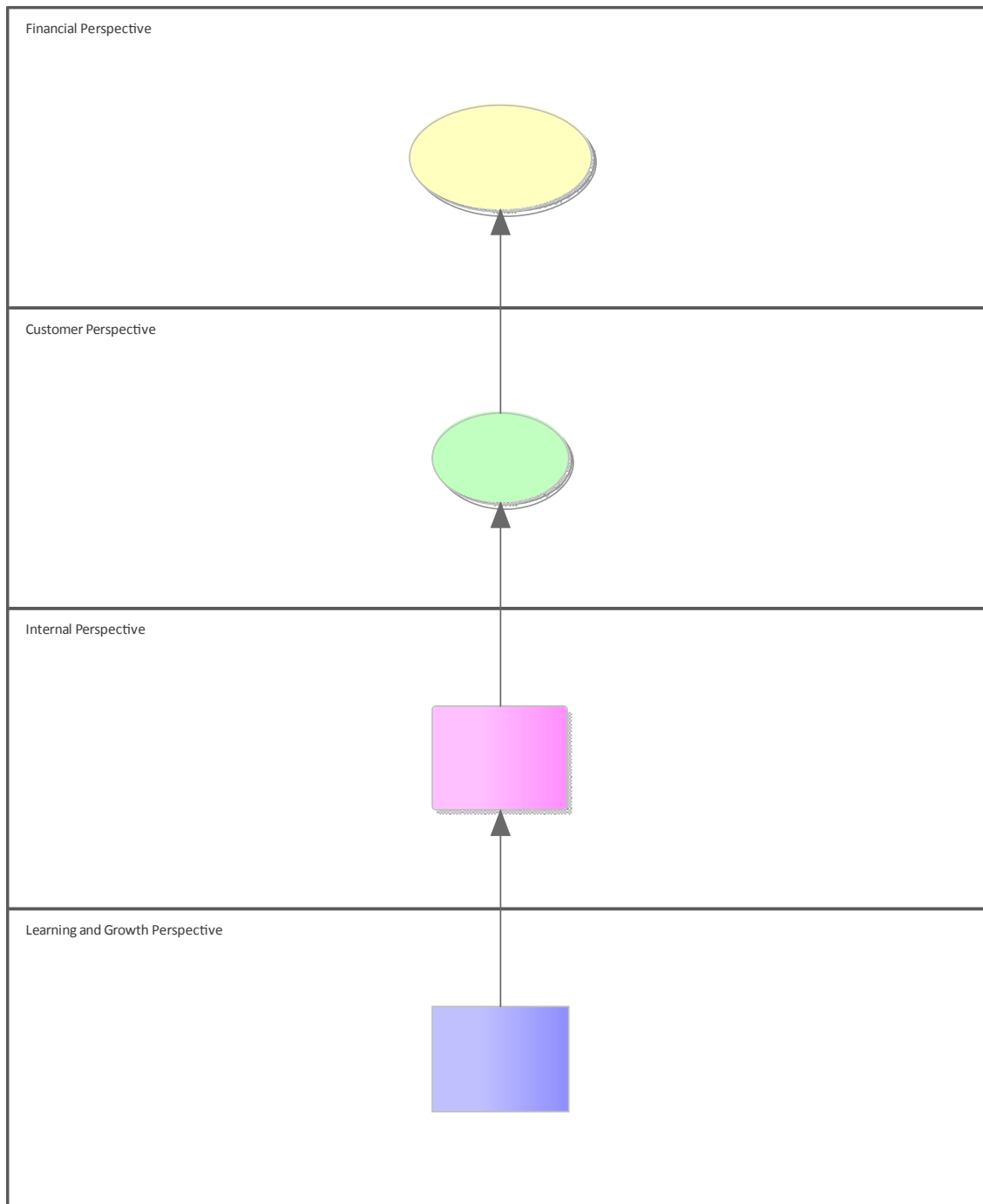
Example Strategy Map (Style 2)

Example Strategy Map (Style 3)

Financial Perspective

Productivity Strategy

Growth Strategy

Long-Term Shareholder Value

Improve Cost Structure

Increase Asset Utilization

Expand Revenue Opportunities

Enhance Customer Value

Customer Perspective

Customer Value Proposition

Price

Quality

Availability

Selection

Functionality

Service

Partnership

Brand

Product/Service
Attributes

Relationship

Image

Internal Perspective

Operations Management Processes	Customer Management Processes	Innovation Processes	Regulatory and Social Processes
Distribution Production Risk Supply	Acquisition Growth Retention Selection	Design/Develop Launch Opportunity ID R&D Portfolio	Community Employment Environment Safety and Health

Learning and Growth Perspective

Human Capital

Information Capital

Organization Capital

Culture

Leadership

Alignment

Teamwork

Mind Mapping

Mind Mapping diagrams offer a flexible, visual notation to convey complex information. Typically they are used for studying, collaborative discussions, problem solving, brainstor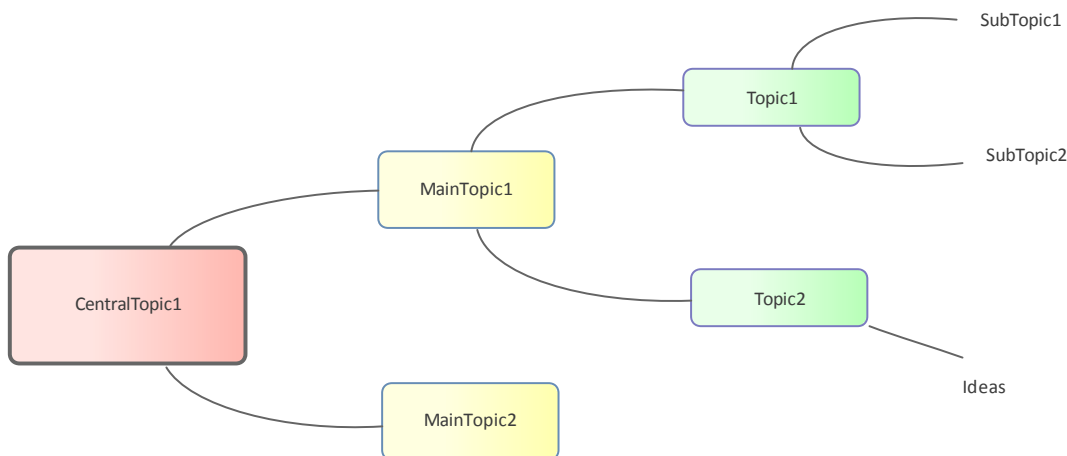ming, presenting complex ideas and decision making. Using this non-linear approach allows the information to be interpreted in many ways, producing unique and innovative solutions.

Create a Mind Mapping Diagram

- Open Enterprise Architect and create a new project file called 'StrategicModel.eap'; create a Simple View entitled 'The Enterprise', containing a Package called 'MindMapping'
- Right-click on the 'MindMapping' Package and select the 'Add Diagram' option; create a new Mind Mapping diagram called 'Brainstorm'
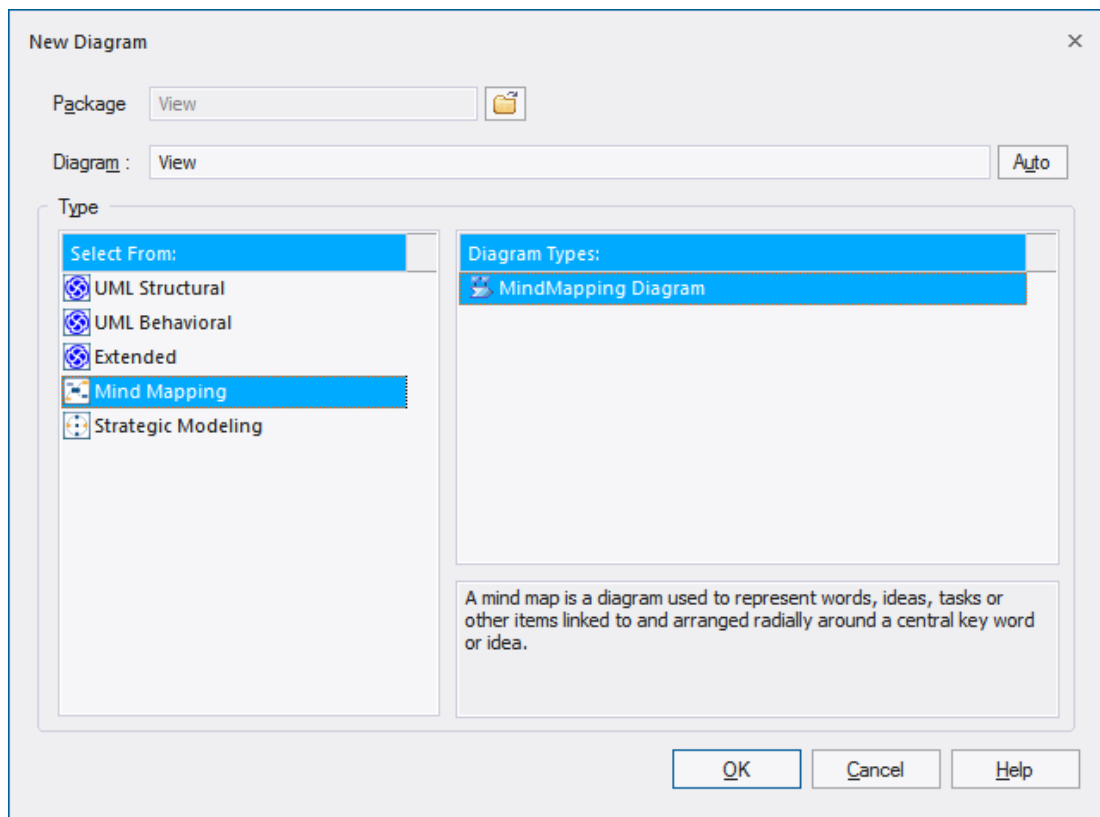
- **Start with the central theme:** Open the Diagram Toolbox and place a Central Topic called 'Benefits of Mind Mapping' in the middle of the Diagram View
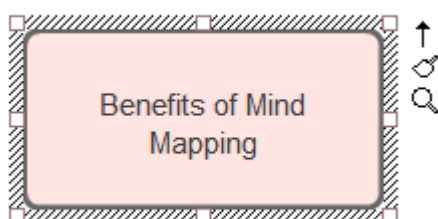
- **Create associated topics:** Use the Quick Linker arrow to create three Main Topics called 'Problem Solving', 'Memory Recall' and 'Decision Making'; alternatively you can use the Relationship connector in the Toolbox to link to related topics

- Select 'Memory Recall' and use the Quick Linker to create three associated Topics called 'Association', 'Visual Memory' and 'The Brain'

- Select 'The Brain' Topic and use the Quick Linker to create four Sub-Topics called 'Emotion', 'Language', 'Imagination', and 'Visual Processing'; your diagram should now contain 11 related topics

- **Create links to on-line resources and important metadata:** Access the 'Properties' dialog for the 'Benefits of Mind Mapping' Central Topic, select the 'Files' tab and enter the URL shown in order to access the Enterprise Architect Help topic *Mind Mapping Diagrams*; Enterprise Architect can easily link to any supplementary material, including local minutes of a meeting, relevant documents, files and web pages
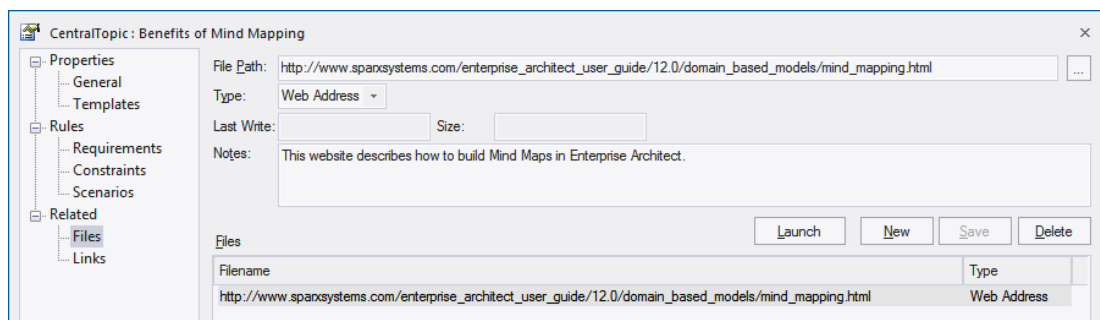
- **Make elements stand out by adding color:** If you want to draw attention to a particular topic or concept, use the 'Layout > Style' ribbon options to alter the font color of the element - for example, change the 'Visual Processing' Sub-Topic to the color blue; color can be used to structure information and enhance readability

- **Layout the diagram:** Select the 'Layout > Tools > Diagram Layout > Open Diagram Layout Tools' option to alter the appearance of a diagram - these tools provide a large number of customization options to enhance the visual layout and presentation of your diagrams; please note, Mind Mapping diagrams traditionally use curved lines
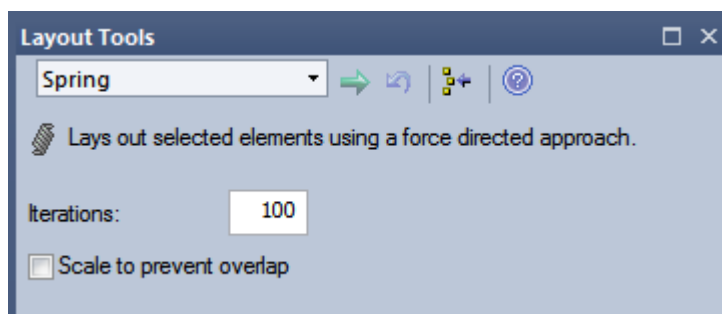
- **Adjust the height and width of your diagram:** Right-click your Mind Mapping diagram to modify the layout of the diagram, based on the options 'Align elements (left, right, top, bottom, centers)', 'Modify size (auto-size, same height and width)', 'Space evenly (Equal distance across and down)' and 'Layout selected elements'

Mind Mapping Example Diagram

Decision Making

Problem Solving

Benefits of Mind Mapping

Association

Visual Memory

Memory Recall

The Brain

Emotion

Language

Imagination

Visual Processing
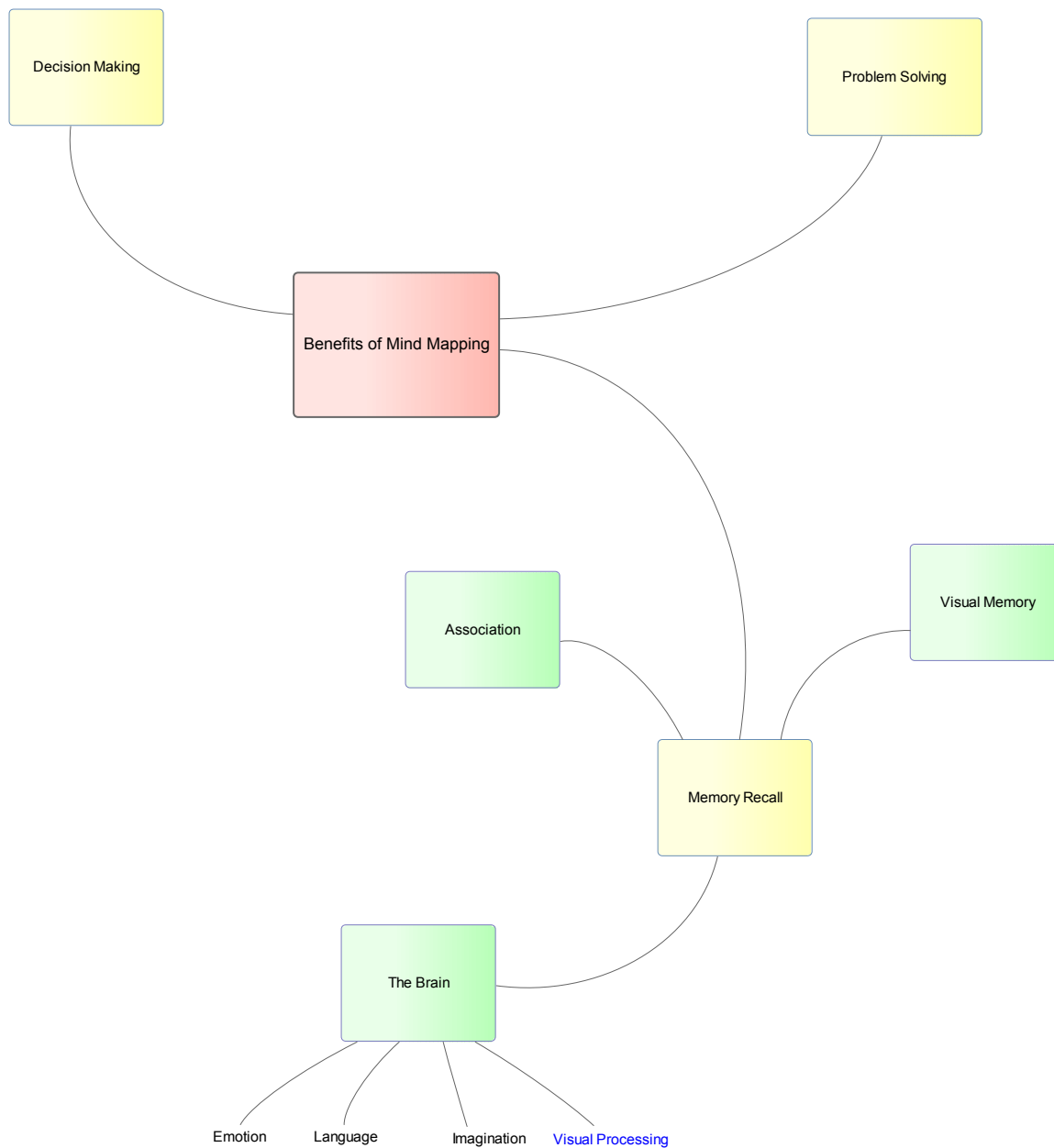
Value Chain

A Value Chain (Porter, 1985) examines the performance and cost of each value-creation activity within an organization. It is a tool that can be used for strategic planning purposes, to identify parts of an organization that provide the best value for customers and shareholders. A Value Chain is often used in conjunction with financial reporting to control cost drivers and identify areas that need improvement.

If a company can derive greater value from one of the activities in the Value Chain, they will be able to beat the competition, creating greater value for the shareholder and customer. For example: A business might have access to cheaper raw materials (Inbound Logistics), use a more efficient manufacturing process (Operations) or reduce transport and storage costs (Outbound Logistics). These primary activities are in turn supported by a number of secondary activities such as HR Management and Technology. A business is able to gain a competitive advantage by analyzing each activity in the chain, benchmarking the activity against the competition, improving processes, providing better differentiation and reducing costs.

Enterprise Architect is able to link each activity in the Value Chain to support documentation, including budgets and benchmark results. The ability to link to competitors' websites and web based market information extends the

power of the Value Chain beyond a traditional text based model.

Creating a Value Chain

- Create a new Package called 'Value Chain'
- Create a new Value Chain diagram called 'Our Company' and click on the OK button
- Select the 'Value Chain' Pattern from the Toolbox and place it on the Diagram View
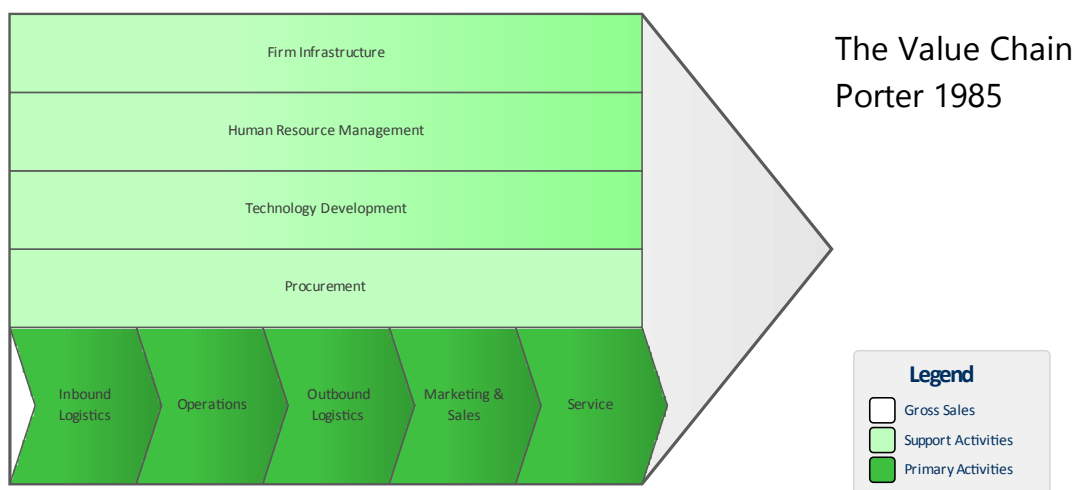
- The default Value Chain can be customized to reflect how your company operates
- You can easily modify the color of each activity using the 'Layout' ribbon, 'Style' panel

Organizational Chart

An Organizational Chart (Org Chart) is a graphical representation of the structure of an organization, describing the roles and responsibilities of each staff member. Enterprise Architect provides an Org Chart Pattern to speed up the creation of Organizational Charts.

When modeling the structure of an organization it is quite common to use an Organizational Chart to describe the structure in terms of relationships between personnel or departments. Each rectangle represents a person, position or department, while each connector represents a hierarchical relationship or command structure. Enterprise Architect can store notes and Linked Document for each position to provide more detail than a traditional Org Chart.

Creating an Organizational Chart

This example illustrates how easy it is to modify the basic Pattern to suit your individual needs.

Create a new Package called 'The Org Chart'. Right-click on the Package and then select 'Add Diagram'. Create a new Organizational Chart called 'Company Structure' and click on the OK button.

From the Toolbox, select the 'Org Chart' Pattern and drag it onto the diagram. Modify the Pattern elements by selecting the ellipsis button next to each role.
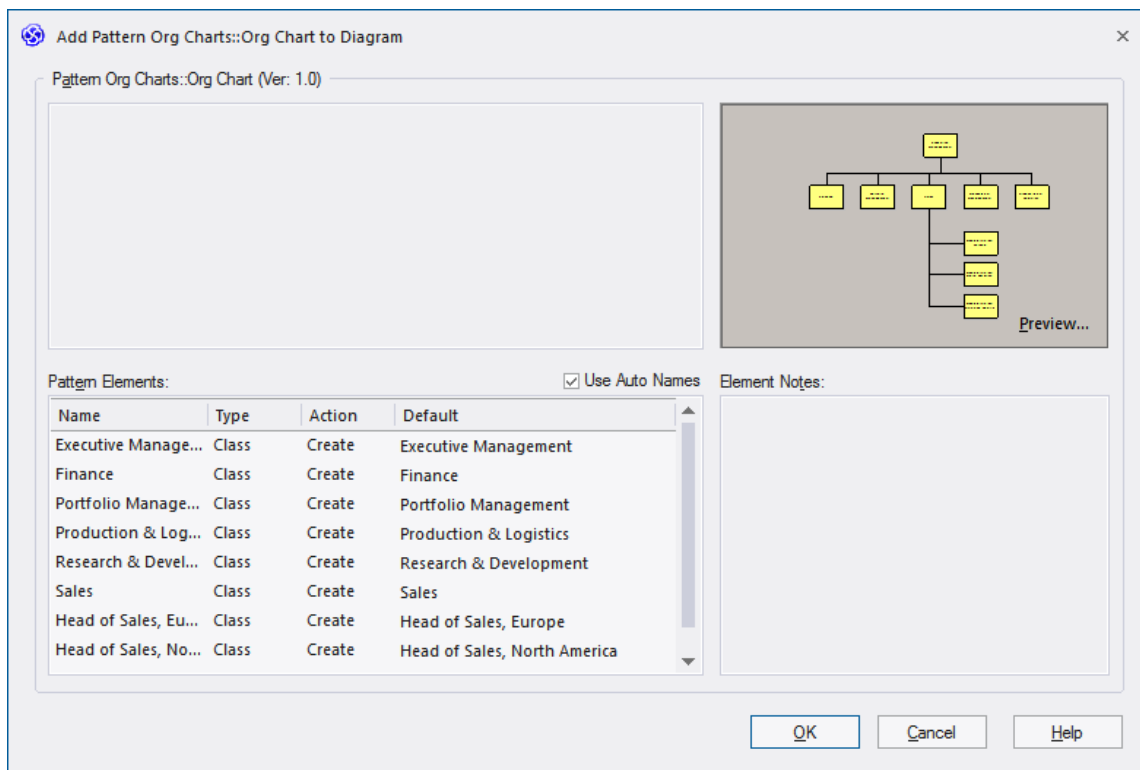
Create these roles in order:

- Marketing Manager
- Sales Manager
- Accounting
- Chief Financial Officer
- Head of Logistics
- Research and Development
- Chief Technical Officer
- Chief Operations Officer
- Chief Executive Office

Move the Sales Manager role so that they can report directly to the CEO.

Delete the connector to the Sales Manager role. Move this

role next to the Chief Operations Officer. Select the 'Reports to (Vertical Tree)' connector from the Toolbox and link Sales Manager to the Chief Executive Officer.

Select the Chief Executive Officer and change the Line and Fill Color to 'Rose' using the 'Layout' ribbon, 'Style' panel.

Organizational Chart - Completed Example

Each Role in this diagram can be linked to a position description, web page or personal CV via the element's 'Properties' dialog or using a Linked Document. Enterprise Architect can also generate RTF and HTML documentation in order to publish the Organizational Chart for distribution to staff.
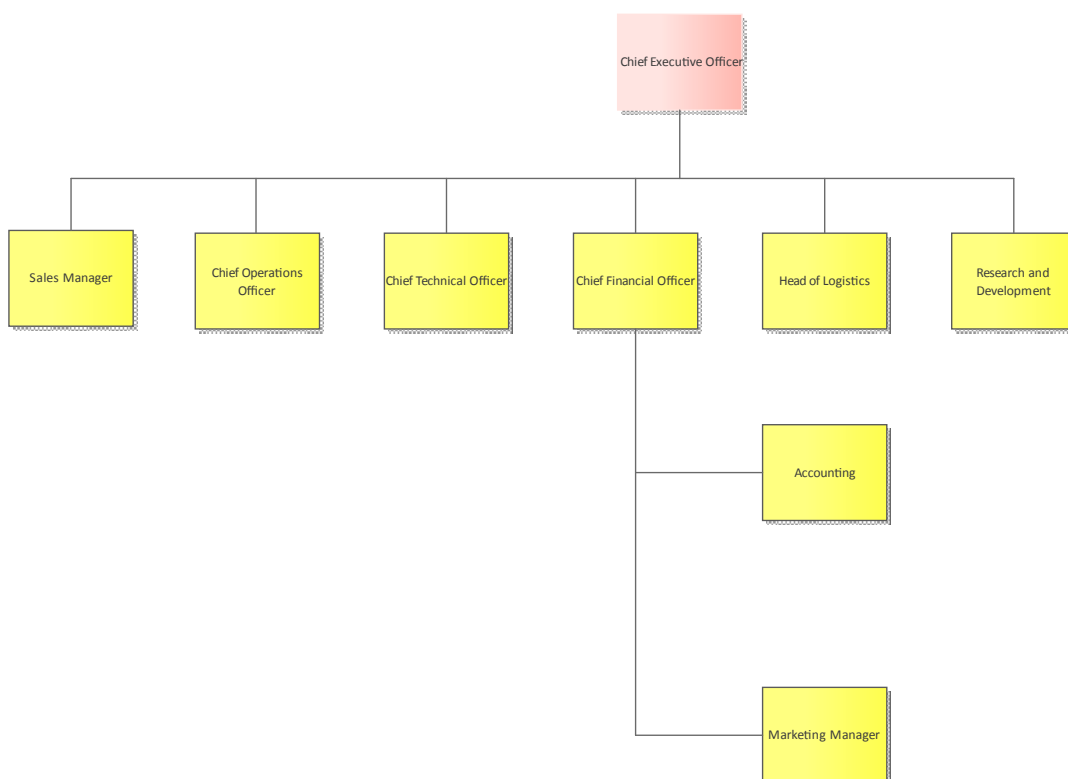
Balanced Scorecard

Traditional performance management tools only focus on financial outcomes and often fail to consider other important factors such as loyal customers or skilled staff. A more balanced approach is to consider non-financial performance measures in relation to Company Strategy. For example, customers might be prepared to pay more for goods and services produced from an environmentally friendly company. A Balanced Scorecard *(Kaplan and Norton, 1992)* gives equal weight to four different perspectives. In a single diagram you can record key financial and non-financial goals, with links to meaningful benchmarks and performance measures. The Scorecard acts as a communication and measurement tool to address strategic planning.

Balanced scorecard - four perspectives:

- Financial
- Customer
- Internal Business Process
- Learning and Growth

Each perspective must consider these items:

- Strategic Objectives (growth, profitability, customer satisfaction, process excellence)
- Measures (revenue growth, return on equity, number of complaints, yield)
- Targets

- Initiatives

Creating a Balanced Scorecard

1. Create a new Package called 'Balanced Scorecard'.
2. Right-click on the 'Balanced Scorecard' Package and select 'Add Diagram'.
3. Create a new Balanced Scorecard diagram called 'Four Perspectives' (enter the name, then click on the 'Hamburger' icon, select 'Strategy > Strategy > Strategic Modeling > Balanced Scorecard', and click on the OK button).
4. In the Toolbox, open the 'Patterns' section and drag the 'Balanced Scorecard' Pattern onto the Diagram View. The 'Add Pattern Balanced Scorecards :: Balanced Scorecard to Diagram' dialog displays.
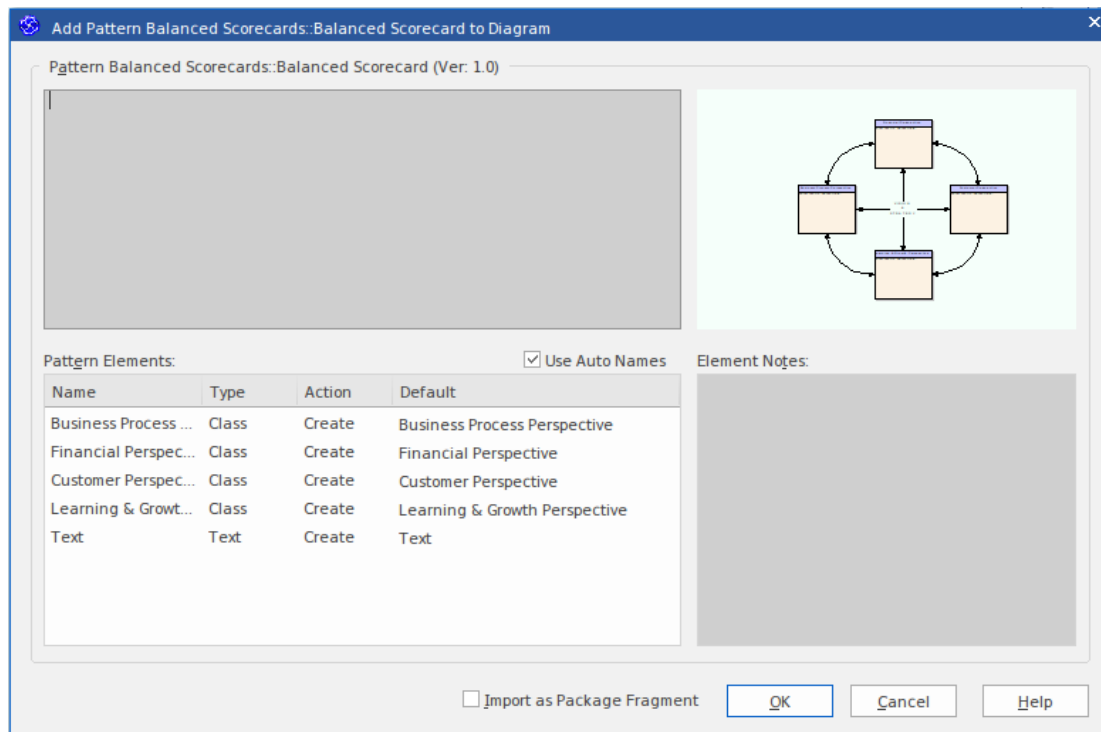
5. Click on the OK button; a confirmatory prompt displays.
   Click on the Yes button to generate the diagram shown
   here.

Financial Perspective

Increased turnover
Reduce input costs
Decrease overtime

Business Process Perspective

Order stock online
Simplify roster management
Staff management - online
Record time sheets - online
Customer loyalty scheme
Elicit customer feedback

**Good Service
Happy Customers
Online Strategy**

Customer Perspective

Excellent customer service
Friendly staff
Customer loyalty scheme
Premium food and drinks

Learning & Growth Perspective

Improve customer service
Promote RSA training
(Responsible serving of alcohol)
Provide access to wage information

Balanced Scorecard Diagram

This diagram illustrates how the company vision and
strategy is directly tied to each of the four perspectives.
Resource allocation is based on results and strategy rather
than a set of arbitrary financial numbers from the previous
financial year. The Balanced Scorecard approach helps to
ensure that business activity is engineered to adhere to
company values and meet targets. Enterprise Architect
allows each perspective to be linked to supporting
documentation, model information and rigorous benchmarks

that verify whether strategic initiatives have been undertaken and associated targets met.

Analysis Models

Analysts have typically had to work in a variety of text-based tools and general purpose drawing packages. The disparate nature of the output from these tools required analysts to copy and paste content between the tools and there was also the considerable task of keeping all the heterogeneous documents up-to-date. Enterprise Architect equips the analyst with a single tool that can seamlessly tie all their work together in a rigorous model that can be updated in a single place, versioned, and used to produce high quality documentation in a variety of formats. The models they create can be traced back to strategic models including stakeholder goals and business drivers and plans, and the same models can also be traced down to architectural solutions and designs creating a single fabric of knowledge.

Analysis Model Diagram Types

Analysis Tool	Detail
Data Flow Diagrams	A Data Flow diagram (DFD) is a graphical representation of the flow of data through an information system; it can also be used to visualize data processing (structured design).

	Developing a DFD helps in identifying the transaction data in the data model.
Custom Models	Custom models provide a number of extensions to the UML model and help you to perform exploratory and non-rigorous experimentation with model elements and diagrams. For example, using a Custom diagram you can model requirements, user interfaces or custom designs.
Mind Mapping	A Mind Map is an image-centered diagram used to represent semantic or other connections between words, ideas, tasks or other items arranged radially around a central key word or idea. A Mind Map is used to generate, visualize, structure and classify ideas, and as an aid in study, organization, problem solving, decision making, and writing.

Mind Mapping

This text is derived from the Mind Map entry in the online Wikipedia:

"A Mind Map is a diagram used to represent words, ideas, tasks or other items linked to and arranged radially around a central key word or idea. It is used to generate, visualize, structure and classify ideas, and as an aid in study, organization, problem solving, decision making, and writing."

"A Mind Map is an image-centered diagram that represents semantic or other connections between portions of information. By presenting these connections in a radial, non-linear graphical manner, it encourages a brainstorming approach to any given organizational task, eliminating the hurdle of initially establishing an intrinsically appropriate or relevant conceptual framework to work within."

"The elements are arranged intuitively according to the importance of the concepts and are organized into groupings, branches, or areas. The uniform graphic formulation of the semantic structure of information on the method of gathering knowledge, may aid recall of existing memories."

The use of the term 'Mind Maps' is trademarked in the UK and the USA by The Buzan Organization, Ltd.

Mind Mapping in Enterprise Architect

Within Enterprise Architect you can develop Mind Maps quickly and simply, through use of an MDG Technology integrated with the Enterprise Architect installer. The Mind Mapping facilities are provided in the form of:
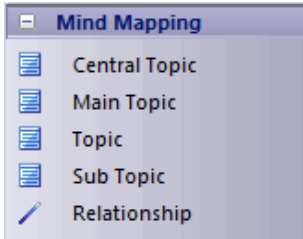
- A Mind Mapping diagram type, accessed through the 'New Diagram' dialog

- A 'Mind Mapping' page in the Toolbox

- Mind Mapping element and relationship entries in the 'Toolbox Shortcut' menu and Quick Linker

Access

Use any of the methods outlined here to display the Diagram Toolbox, then click on 🔍 to display the 'Find Toolbox Item' dialog and specify 'Mind Mapping'.

Ribbon	Design > Diagram > Toolbox
Keyboard Shortcuts	Ctrl+Shift+3
Other	Click the » icon on the Diagram caption bar to display the Diagram Toolbox

# Mind Mapping Toolbox Page

Page Appearance	Description
	• Central Topic is the main theme of the Mind Map; you would normally have one or two of these on the diagram, but can add as many as are necessary • Main Topic represents the immediate concepts generated by the Central Topic • Topic represents the larger divisions of a Main Topic • Sub Topic represents the finer divisions of a Topic or Main Topic; you could also have Subtopics of Subtopics to represent increasingly finer distinctions • Relationship represents the connection between any two elements; you can have several Relationships per element Each relationship has three anchor points, so you can curve the lines to develop the flow of concepts more easily

Mind Map Diagram

This is the general appearance of a Mind Mapping diagram.
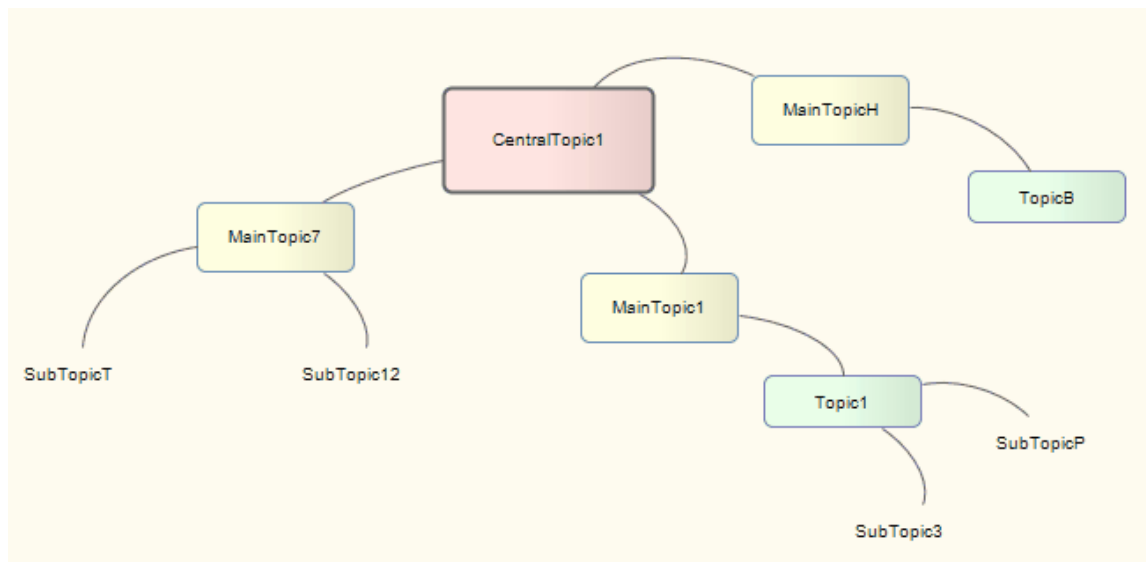
As the elements can represent any concept, object or relationship, you can use the full range of element properties and features to expand on what the element represents, including adding Note elements. However, to preserve the simplicity and readability of the diagram itself, you cannot display the element compartments on the diagram.

Whiteboards

The Whiteboard Technology provides a set of Diagram Toolbox pages with a selection of stereotyped element icons that generate simple shapes suitable for use on hand-drawn and whiteboard diagrams.

Access

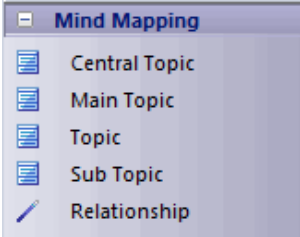Use any of the methods outlined here to display the Diagram Toolbox, then click on to display the 'Find Toolbox Item' dialog and specify 'Whiteboard'.

Ribbon	Design > Diagram > Toolbox
Keyboard Shortcuts	Ctrl+Shift+3
Other	Click the »icon on the Diagram caption bar to display the Diagram Toolbox

Whiteboard Toolbox Page

Page	Description

Appearance	
Mind Mapping Central Topic Main Topic Topic Sub Topic Relationship	The Whiteboard Toolbox pages provide these shape icons: • Shapes: Circle, Oval, Square, Rectangle, Round Rectangle, Triangle, Diamond, Cloud, Cube, Cyclinder, Vertical Line, Horizontal Line, Flag, Star, Home, Clock, Tag, Callout and Text Element. • Arrows: Up, Down, Left and Right • Thin Arrows: Up, Down, Left and Right • Lines: Line, Directed Line, Dashed Line, Directed Dashed Line, Curve, Directed Curve.

Whiteboard Diagram

You can create either a Whiteboard diagram or a Hand-drawn diagram. Both of these diagrams, when opened, cause the Whiteboard Toolbox pages to be opened.

Custom Diagram

A Custom diagram is an extended Class diagram that is used to capture requirements, user interfaces or custom-design models.

Custom models provide a few extensions to the UML model and enable some exploratory and non-rigorous experimentation with model elements and diagrams. You generate Custom diagram elements and connectors from the 'Custom' pages of the Diagram Toolbox.

Custom Diagram Element Toolbox Icons

Icon	Description
Package	Packages are used to organize your project contents, but when added onto a diagram they can be use for structural or relational depictions.
Requirement	A Requirement element captures the details of a system requirement.
Issue	An Issue element represents an item of concern (that might or might not occur) or a failure to meet a defined requirement in the current system.

Change	A Change element represents a change in the defined requirements for the current system and can be used to request and manage the change process.
Screen	A Screen element is used to prototype a User Interface screen flow.
UI Control	A UI Control element represents a user interface control element (such as an edit box).
Test Case	A Test Case is a stereotyped Use Case element which enables you to give greater visibility to tests.
Entity	An Entity is a stereotyped Object that models a store or persistence mechanism that captures the information or knowledge in a system.

Custom Diagram Connector Toolbox Icons

Icon	Description

Associate	An Association implies that two model elements have a relationship, usually implemented as an instance variable in one or both Classes.
Aggregate	An Aggregation connector is a type of association that shows that an element contains or is composed of other elements.
Generalize	A Generalization is used to indicate inheritance.
Realize	A Realizes connector represents that the source object implements or Realizes its destination object.
Nesting	A Nesting Connector is an alternative graphical notation for expressing containment or nesting of elements within other elements.

Data Flow Diagrams

This text is derived from the Data Flow diagram entry in the online Wikipedia.

'A Data Flow diagram (DFD) is a graphical representation of the 'flow' of data through an information system. A Data Flow diagram can also be used for the visualization of data processing (structured design). It is common practice for a designer to draw a context-level DFD first which shows the interaction between the system and outside entities. This context-level DFD is then 'exploded' to show more detail of the system being modeled.'

'Data Flow diagrams were invented by Larry Constantine ... based on Martin and Estrin's "data flow graph" model of computation. (They) are one of the three essential perspectives of Structured Systems Analysis and Design Method SSADM. The sponsor of a project and the end users will need to be briefed and consulted throughout all stages of a system's evolution. With a Data Flow diagram, users are able to visualize how the system will operate, what the system will accomplish, and how the system will be implemented. The old system's Data Flow diagrams can be drawn up and compared with the new system's Data Flow diagrams to draw comparisons to implement a more efficient system.'

'Developing a DFD helps in identifying the transaction data in the data model.'

Access

Use any of the methods outlined here to display the Diagram Toolbox, then click on 🔍 to display the 'Find Toolbox Item' dialog and specify 'Data Flow Diagrams'.

Ribbon	Design > Diagram > Toolbox
Keyboard Shortcuts	Ctrl+Shift+3
Other	Click the » icon on the Diagram caption bar to display the Diagram Toolbox
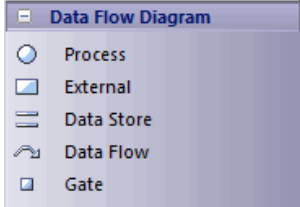
Data Flow Diagrams in Enterprise Architect

Within Enterprise Architect, you can develop Data Flow diagrams quickly and simply through use of an MDG Technology integrated with the Enterprise Architect installer. The Data Flow diagram facilities are provided in the form of:

· A Data Flow diagram type, accessed through the 'New Diagram' dialog

· A 'Data Flow Diagram' page in the Diagram Toolbox

· Data Flow element and relationship entries in the Toolbox

'Shortcut Menu' and Quick Linker

Data Flow Diagram Toolbox Page

Page Appearance	Description
Data Flow Diagram toolbox: Process External Data Store Data Flow Gate	Process is a process or activity in which data is used or generated. External represents an external source, user or repository of the data. Data Store represents an internal physical or electronic repository of data, into and out of which data is stored and retrieved. Data Flow (connector) represents how data flows through the system, in physical or electronic form. Gate represents the termination point of incoming and outgoing messages on a lower level diagram (that is, messages to and from processes depicted elsewhere).

Data Flow Diagram Appearance

When dragged onto a Data Flow diagram, the elements and relationships have these appearances:
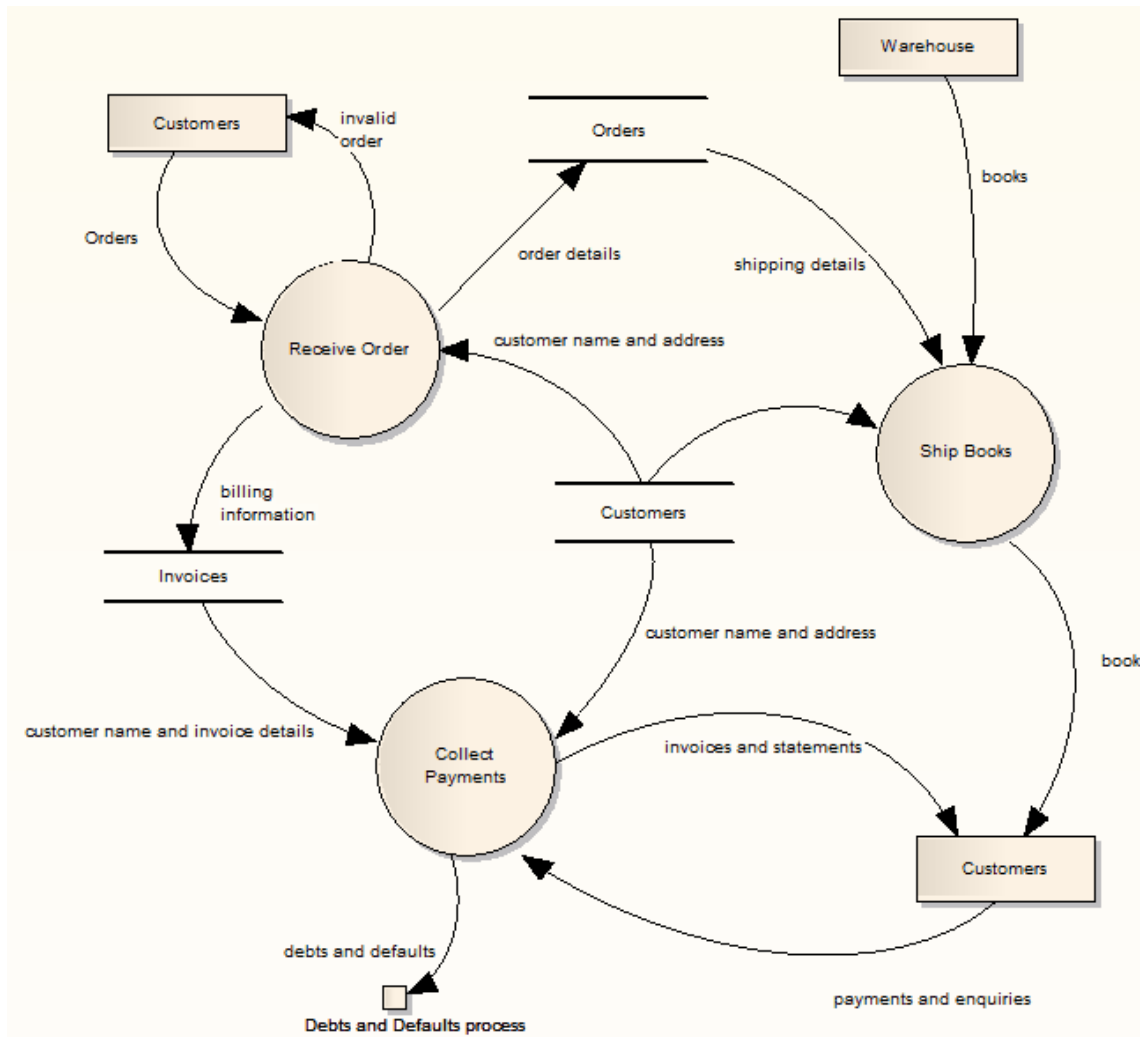
To preserve the simplicity and readability of the diagram, you cannot display element compartments on the diagram.

Context Diagram

A Context diagram is a top-level Data Flow diagram that has just one Process element representing the system being

modeled, showing its relationship to external systems.

Analysis Stereotypes

Enterprise Architect provides some built in stereotypes that you can assign to an element during analysis. The effect of each of these stereotypes is to display a different icon from the normal element icon, giving a visual key to the element's purpose. The main types of inbuilt icon for elements created using the stereotypes include:

- Boundary - for a system boundary (for example, a Login screen)
- Control - to specify an element is a controller of some process (as in the Model-View-Controller Pattern)
- Entity - the element is a persistent or data element

The elements created using these stereotypes are illustrated in this Robustness diagram. Also see the Business Modeling elements, used in Business Modeling diagrams and Business Interaction diagrams.

Example of Analysis Stereotypes in use

Use the boundary, control and entity stereotypes to alter the appearance and meaning of Objects or Classes. A boundary is a user interface (screen); a control is an active or managing component; and an entity is a persistent element (the data)
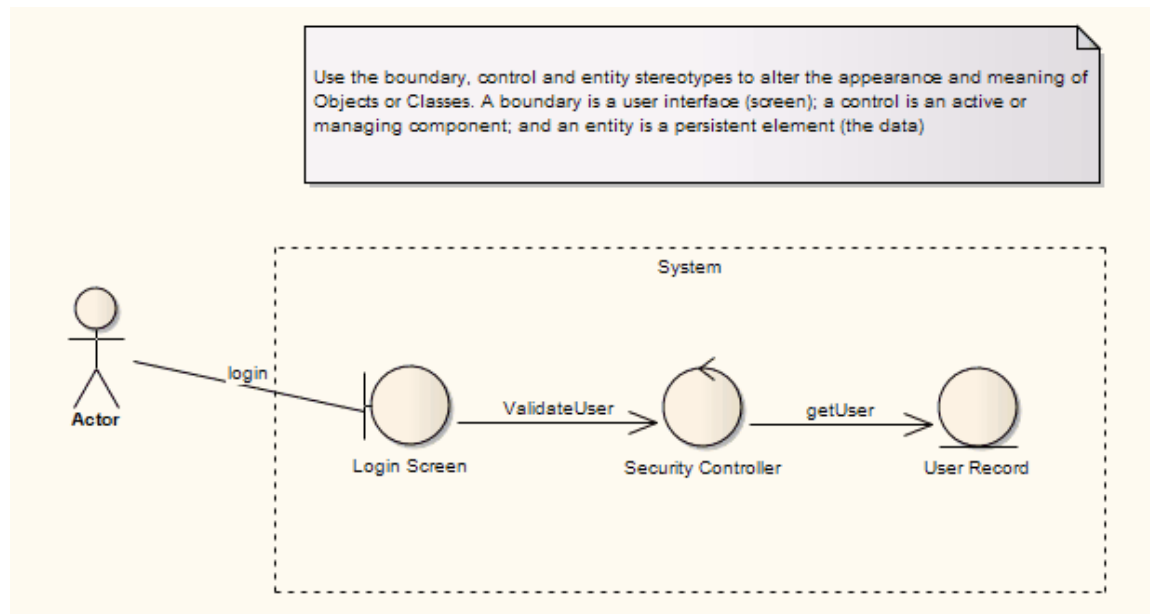
Analysis Diagram

An Analysis diagram is a simplified Activity diagram, used to capture high level business processes and early models of system behavior and elements. It is less formal than some other diagrams, but provides a useful means of capturing the essential business characteristics and requirements. Robustness diagrams, used extensively in ICONIX, can be created as Analysis diagrams.

You generate Analysis diagram elements and connectors from the 'Analysis' pages of the Diagram Toolbox.

Example Diagram

[Example Analysis Diagram](#)

Analysis Diagram Element Toolbox Icons

Icon	Description
Actor	An Actor is a user of the system; 'user' can mean a human user, a machine, or even another system or subsystem in the model.
Object	

	An Object is a particular instance of a Class at run time.
⟫ Process	A Process is an Activity element with the stereotype process, which expresses the concept of a business process.
◌ Collaboration	A Collaboration defines a set of cooperating roles and their connectors.
◌ Collaboration Use	A Collaboration Use element allows for a Pattern defined by a Collaboration to applied to a specific situation.
▷ Send	The Send element is used to depict the action of sending a signal.
▷ Receive	A Receive element is used to define the acceptance or receipt of a request.
▣ Information Item	An Information Item element represents an abstraction of data, which data can be conveyed between two objects.
◇ Decision	A Decision is an element that indicates a point of conditional progression: if a condition is true, then processing continues one way; if not, then another.

◈ Merge	A Merge Node brings together a number of alternative flow paths in Activity, Analysis and Interaction Overview diagrams.
⊢◯ Boundary	A Boundary is a stereotyped Object that models some system boundary, typically a user interface screen.
⊙ Control	A Control element represents a controlling entity or manager that organizes and schedules other activities and elements.
◯ Entity	An Entity is a stereotyped Object that models a store or persistence mechanism that captures the information or knowledge in a system.

Analysis Diagram Connector Toolbox Icons

Icon	Description
Information Flow	An Information Flow represents the flow of Information Items (either Information Item elements or classifiers) between two

	elements in any diagram.
Object Flow	An Object Flow connects two elements, with specific data passing through it, modeling an active transition.
Associate	An Association implies that two model elements have a relationship, usually implemented as an instance variable in one or both Classes.
Realize	A Realizes connector represents that the source object implements or Realizes its destination object.
Representation	The Representation relationship is a specialization of a Dependency, connecting Information Item elements that represent the same idea across models.

Example Analysis Diagram

Enterprise Architect supports some of the Eriksson-Penker Business Extensions that facilitate business process modeling. You can also load the complete Eriksson-Penker Business Extensions UML Profile into Enterprise Architect to create detailed process models.
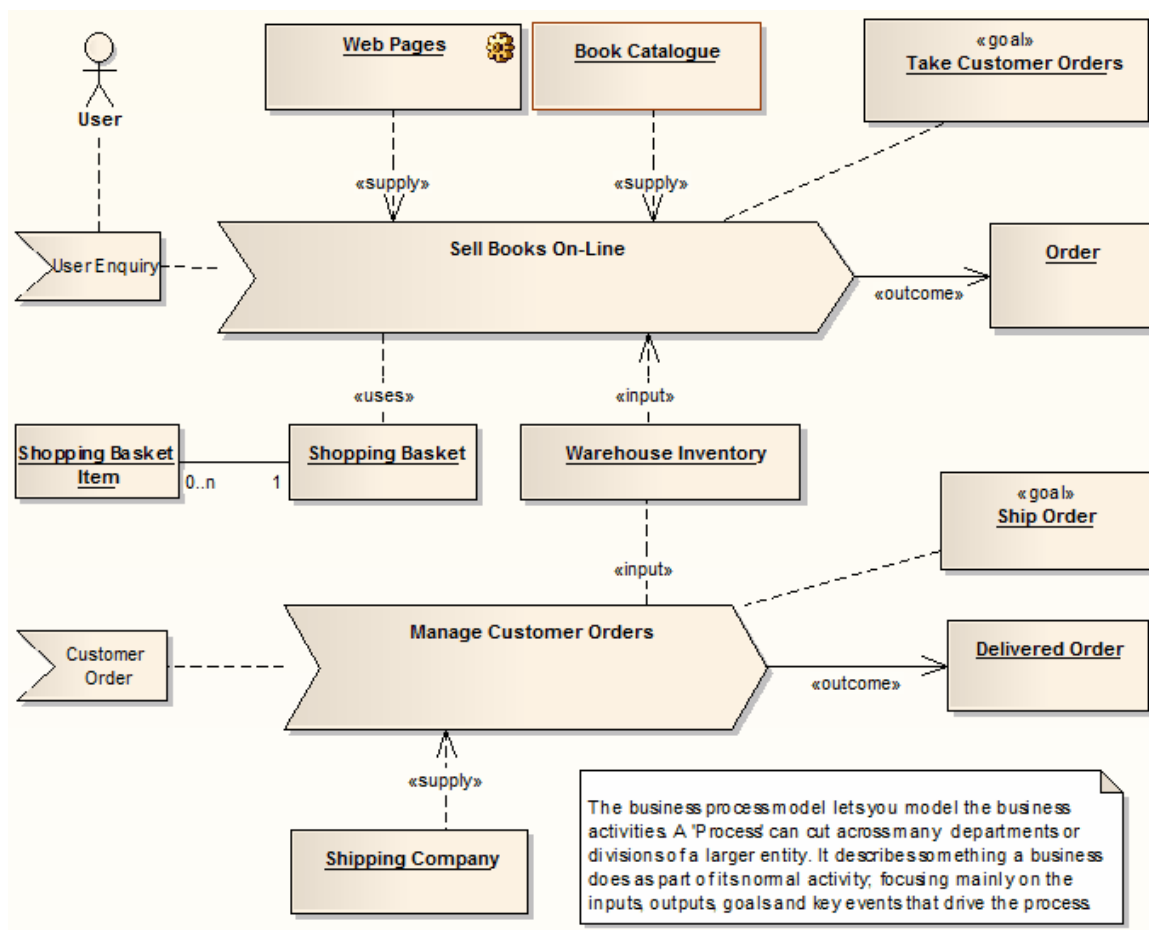
Flow Chart

A Flow Chart is a graphical representation of a sequence of events, helping decision makers understand the relationship between their decisions and a given outcome. Flow Charts use a range of simple geometric shapes to represent a process, decision, storage or output.

- Begin
- End
- Process
- Defined Process
- Input/Output
- Storage
- Decision
- On Page Connector
- Off Page Connector
- Display
- Report
- Annotation
- Connector
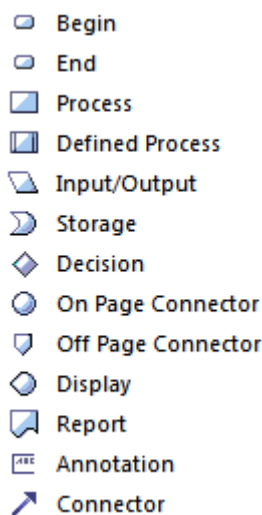
Creating a Flow Chart

Create a flowchart for hiring staff at a Restaurant.

1. Create a new Package called 'Flow Chart'. Add a Flow Chart diagram from the Strategic Modeling category.

2. Place a 'Begin' and 'End' icon onto the diagram. Name the two respective items 'Begin Selection Process' and 'End HR Process'.

3. Create two different Processes called 'Advertise Position' and 'Review Written Application'.

4. Create a Decision called 'Met Due Date'. This decision will be used to determine if a candidate will proceed to the written application stage.

5. Draw a connector between 'Begin Selection Process' and 'Advertise Position' followed by another connector between 'Advertise Position' and 'Met Due Date'.

6. Right-click on the Connector located between 'Advertise Position' and 'Met Due Date' and select the ControlFlow properties. Name the Connector 'Yes'. By labeling the Connector with the word 'Yes' we assert that if an application is received by the due date then it will be reviewed by staff in the Human Resources department. Any submissions received after the due date will not be reviewed by staff.

7. Create a Decision called 'Address Selection Criteria'. Draw a Connector from 'Review Written Application' to the Decision 'Address Selection Criteria'.

8. Create a Process called Interview Applicant'. Draw a Connector from 'Address Selection Criteria' to 'Interview Applicant' and label the Connector 'Yes' denoting that if the applicant has addressed the selection criteria then proceed to 'Interview Applicant'.

9. Create a Decision called 'Appointment Made'. For the 'Yes' response, draw a Connector to 'End HR Process'.

10. For the 'No' response, draw a connector to 'Advertise Position'. Note: you could use a Bezier Connector (a curved line) to improve the diagram layout.

11. Select the entire diagram (Ctrl+A) and select 'Align

Centers | Align Vertically'.

Begin Selection
Process

Advertise Postion

Met Due Date

Yes

Review Written
Application

Address
Selection
Criteria

No

Yes

Interview Applicant

Appointment
Made

Yes

End HR
Process
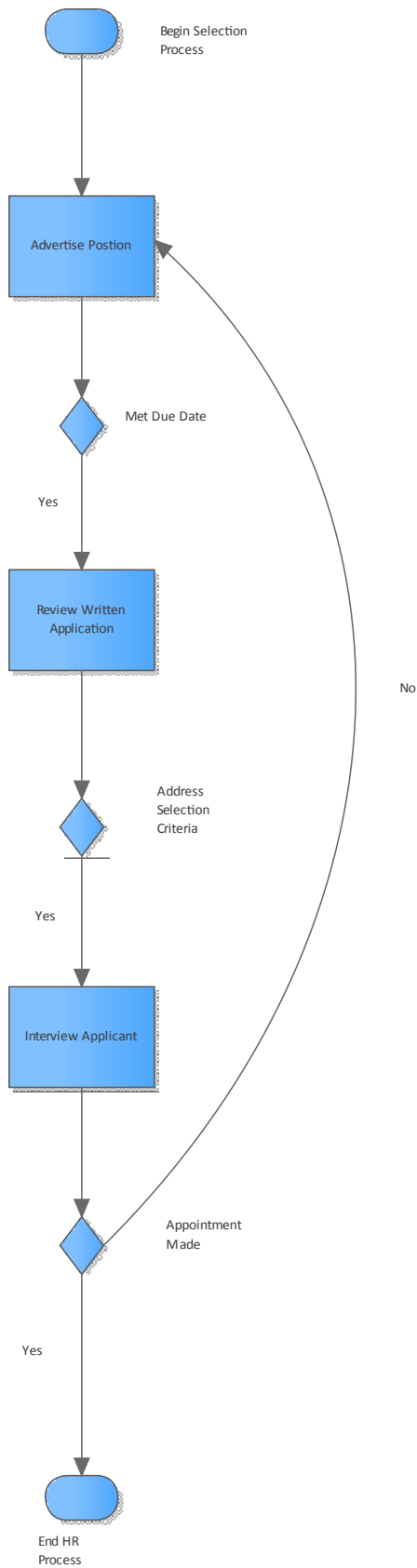
Requirements

Requirements Engineering is the discipline of eliciting, analyzing, specifying and managing requirements. A requirement is essentially a definition of a property that a system or process must be able to perform. Enterprise Architect provides tools and functionality that will assist with all aspects of requirements engineering from elicitation through to requirements management.

Requirement engineering is performed differently depending on the development method being used; traditional processes such as Water Fall will prescribe that the requirements are elicited and analyzed before development work is started, whereas when iterative and incremental methods are used, including Agile, the requirements are incrementally elicited and analyzed.

Requirements also exist at different levels in a process; for example, there are high level requirements such as stakeholder needs, and low level requirements that define how a system component must function. Detailed requirements can be organized into a hierarchy culminating in a high-level requirement, so that satisfying each of the detailed requirements results in meeting the higher-level requirements and ultimately the top-level requirement. This hierarchical structure helps manage the complexity of large systems, with thousands of requirements and many processes being developed to implement those requirements.

Enterprise Architect was one of the first UML tools to

support requirements engineering, and it continues to be a leader in this field with a powerful and versatile feature set supporting all aspects of requirements engineering, including the text based Specification Manager, hierarchical representations, Requirements diagrams for visualizing, automatic documentation generation and requirements management features. Requirements can be given a status that can be conveniently displayed using color codes on a diagram, helping the requirements manager and others to gain a quick visual overview of the status of the requirements.

Gathering Requirements

Gathering requirements is typically the first step in developing a solution, be it for developing either a system or a process. Requirements are gathered from all parties expected to use, maintain or benefit from the solution, and are organized into groups, functional areas and hierarchies as necessary. They can be transcribed into a spreadsheet or a requirements gathering or management tool, or they can be created within an integrated modeling tool such as Enterprise Architect.

The management of requirements is one of the more problematic disciplines in software development, for reasons such as:

- Diverse group input into the requirements
- Organizational boundary divisions

- Tool boundary divisions
- Volatility of requirements
- Imprecision and ambiguity in natural languages

These can cause issues with:

- Traceability and
- Integration with change and configuration management systems

Enterprise Architect can reduce or eliminate these problems in Requirements Management.

Requirements Management and Enterprise Architect

Enterprise Architect is one of the few UML tools that integrate Requirements Management with other software development disciplines in the core product, by defining requirements within the model. Within Enterprise Architect, you can:

- Create and view requirements as entities and properties directly in the model, as simple text descriptions or as diagrammatic representations of the elements and their organization
- Collate the requirements in an external CSV file and then import them into your model
- Detail Use Cases and scenarios directly in the model

- Enter standard attributes (properties) for each Requirement, such as difficulty, status and type, and define your own attributes (properties)

- Trace requirements to Use Cases, Business Rules, Test Cases and analysis artifacts (using, for example, the Relationship Matrix)

- Trace and view the impact of changes on requirements (through, for example, the Traceability window) and review the changes themselves

- Create customer-quality MS Word and HTML reports on requirements

Notes

- All of these features are illustrated by examples in the EAExample.eap model, provided as part of your Enterprise Architect installation in the Enterprise Architect Program Files directory: ...\Program Files\Sparx Systems\EA

- If your project team are not using Enterprise Architect to manage Requirements, they can still access, use and work with them via the Cloud, from an Open Services for Lifecycle Collaboration (OSLC) client tool

Requirements

Enterprise Architect supports requirements definition for enterprise, business, software, hardware and system engineering projects, including functional and non-functional requirements. There are a number of requirement types built into the core product and new types can be added to suit any project. The UML does not formally define a Requirement element, but Enterprise Architect extends the language to provide an element that can be added directly into the repository or through the text based Specification Manager, or created on diagrams. In Enterprise Architect the requirement is treated as a first-class modeling element, and is able to participate in relationships allowing traceability to be established and a project manager or business analyst can track that a project is being designed, built and tested according to the stakeholders needs and its specifications. Use Cases can also be defined and a sophisticated editor helps you to define Scenarios that can be automatically generated to behavior diagrams, enabling the Requirements Analyst to trace to individual steps in a scenario. Using Stereotypes user stories can be modeled, and an Agile Backlog can be defined using the Feature element; as prioritization occurs these can be elaborated into well articulated requirements ready for the development team.

Description

As an analysis step, often it is desirable to capture simple system requirements. These are eventually realized by Use Cases.

In the initial requirement gathering phase, cataloging requirements can be achieved using the Requirement extension on a Custom diagram.

Examples

Requirements can be aggregated to create a hierarchy, as illustrated by this diagram.
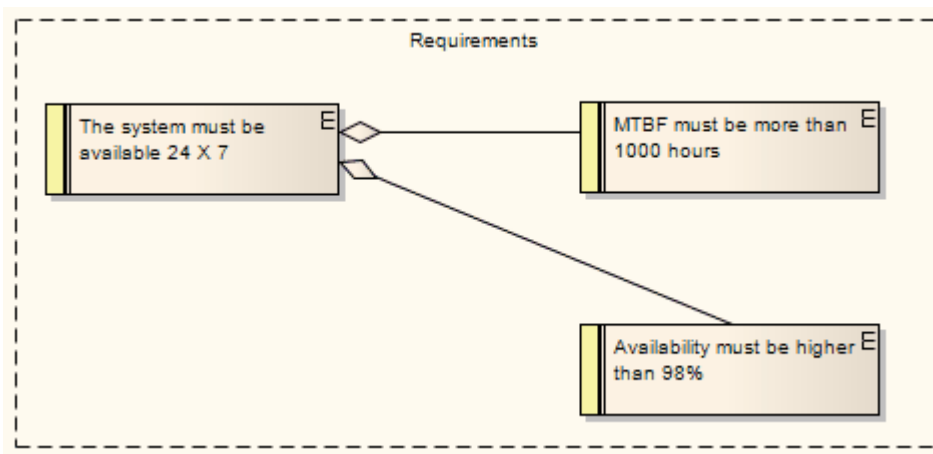
In the next diagram, a requirement that a user can log into a website is implemented by the Login Use Case, which in turn is implemented by the Business Logic, ASP Pages and Login Web Page constructions. Using this approach, you can easily model quite detailed and complex dependencies and implementation relationships.
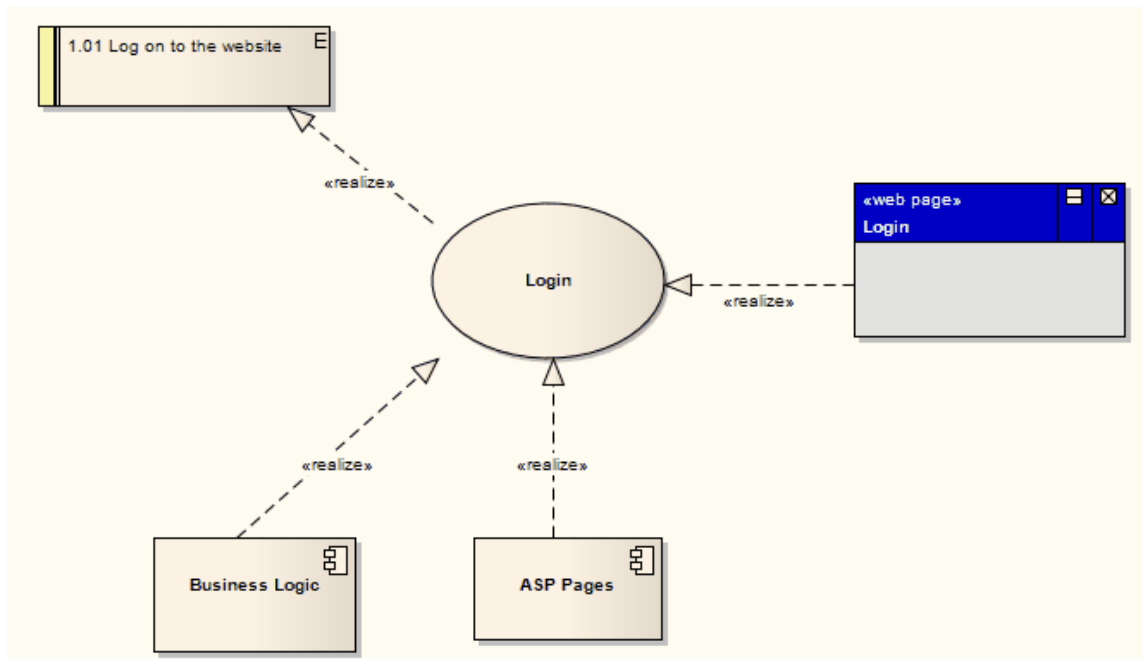
Notes

- External requirements can be displayed with or without an identifying 'E' (for External) in the top right corner of the element; to toggle the display of this letter, select or deselect the 'Show stereotype icon for requirements' checkbox on the 'Preferences' dialog, 'Objects' page

- The colors on Requirement elements identify the status of the requirement; you change the status - and hence color - on the element 'Properties' dialog, and set the color for each status on the 'Status Types' dialog

Toolbox icon
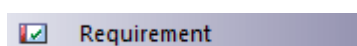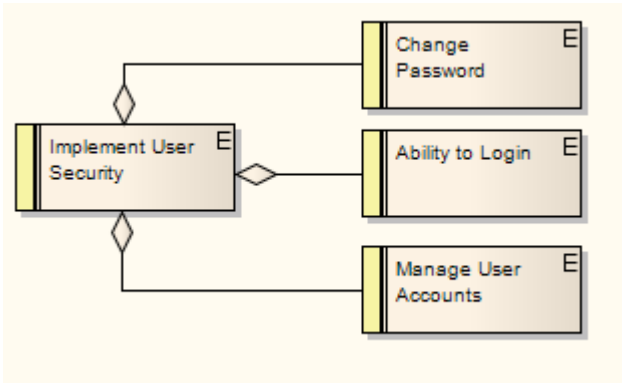
Model Requirements

Requirements are fundamental to the success of any project, regardless of the process being used. Many organizations have traditionally stored and managed their requirements in text tools such as spreadsheets and word processors, but these requirements often remain hidden from the development process. Using Enterprise Architect's model based approach to requirements engineering turns requirements into first class modeling elements. They can be displayed on diagrams and related to the stakeholders that own them, and traces can be created connecting them to other down-process model elements such as Use Cases and application Components.

Properties such as status, phase, complexity and difficulty can be assigned to each requirement, which helps you to manage them easily.

Features

Feature	Detail
Represent Requirements	In Enterprise Architect, a requirement can be modeled as an: • External Requirement - an expectation of the system or process, what the system or process must provide,

	modeled as an *element*; for example, a business requirement or a stakeholder request - Requirements at this level have their own properties and are reported on separately in document reports · Internal requirement – a requirement of an existing element, what the element must do or accomplish, defined as a *property* of the element Requirements Management in Enterprise Architect is primarily concerned with Requirement elements and the elements that implement or realize them.
Requirements in the Model	Requirement elements can be grouped and organized within Requirements diagrams. The Requirement elements are connected to each other by Aggregate relationships to form a hierarchy: It is quite usual to develop a Package of many hundreds of Requirement elements,

	arranged individually and in hierarchies of varying complexity. You can select a Package and use the 'Design > Model > Manage > Level Numbering' ribbon option to highlight the order and arrangement of the Requirements quickly and easily.
	This illustration shows a number of Requirements in a Package, where Level Numbering makes the order and arrangement clear:
	If elements are added, moved or deleted from the Package, the numbering automatically adjusts.
	This numbering can also be applied in a document.
Use Cases	Requirements are implemented (realized) by model elements such as Use Cases, Classes, Interfaces and Components. There are many ways to trace either the Requirement for the feature or service
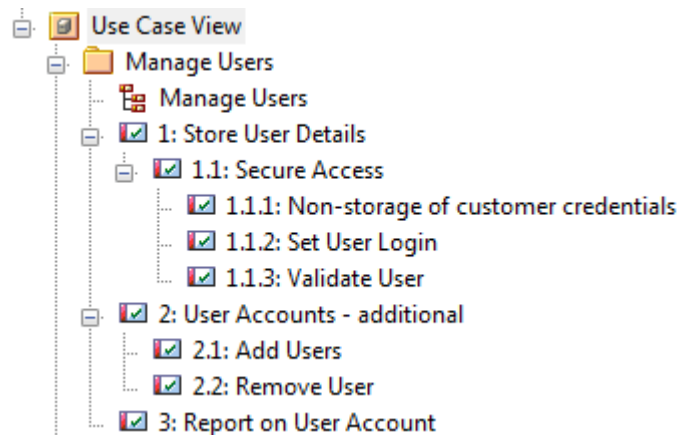
modeled by the elements, or the elements that develop the requirement, most visibly in Traceability diagrams that depict the Requirements and the model elements connected by Realize relationships. The Realize connector enables members of the project team to keep design objectives and development in tandem, and the development path and purpose clear.
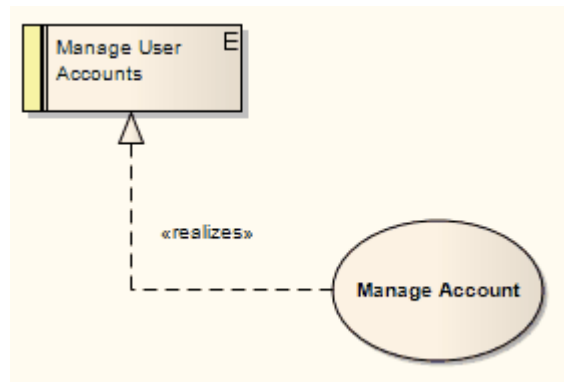
The more usual realization relationship is between a Requirement and a Use Case. A Requirement can be realized by one or more Use Cases, and a Use Case can realize one or more Requirements.

Whilst a Requirement defines a condition that must be met, the Use Case is the key to defining and visualizing how that condition is met. A Use Case diagram depicts the logical grouping of actions, processes and components to achieve a

	required result, and through the use of Actor elements also defines the user and/or system roles participating in the process.
	Each Use Case (as a composite element) can contain a combination of child diagrams that define in greater detail how a particular activity or facility might be implemented - such diagrams include Sequence, Communication, Activity, StateMachine and Business Rule Flow diagrams. The actual implementation of each Use Case is realized by Class, Component and Interface elements organized in their own diagrams. These realizations can also be captured and viewed in Traceability diagrams, depicting the full development pathway from initial requirement through to testing and production.

Requirements Diagram

A Requirements diagram is a Custom diagram used to describe a system's requirements or features as a visual model. Each Requirement is defined as a Requirement element (a Custom element of type Requirement). The actual Requirement, as a text explanation, is the element name (short) or description (long) in the element properties.

Requirement elements can have relationships with other elements, such as other Requirements, Use Cases and Components, to illustrate how a requirement is satisfied by modeling and development. You can track the development arising from a specification or requirement using the Traceability window.

Example Diagram

[Example Requirements Diagram](#)

Requirements Diagram Elements

You can create Requirements diagram elements by dragging them onto the diagram from the 'Requirements' pages of the Diagram Toolbox.

Toolbar Icon	Description

Package	Packages are used to organize your project contents, but when added onto a diagram they can be use for structural or relational depictions.
Requirement	A Requirement element captures the details of a system requirement.
Feature	A Feature is a small, granular function or characteristic expressed in client-valued terms as a satisfaction of a requirement.
Object	An Object is a particular instance of a Class at run time.

Requirements Diagram Connectors

You can create Requirements diagram connectors by dragging them onto the diagram from the 'Requirements' pages of the Diagram Toolbox.

Toolbar Icon	Description
Aggregate	An Aggregation connector is a type of association that shows that an element contains or is composed of other elements.

↗ Inheritance	A Generalization is used to indicate inheritance.
╱ Associate	An Association implies that two model elements have a relationship, usually implemented as an instance variable in one or both Classes.
⤢ Implements	A Realizes connector represents that the source object implements or Realizes its destination object

Example Requirements Diagram

These examples illustrate possible structures of a Requirements diagram. Use Cases and Components in the system can be linked back to the Requirement elements to define how a particular system requirement is met.

Example 1
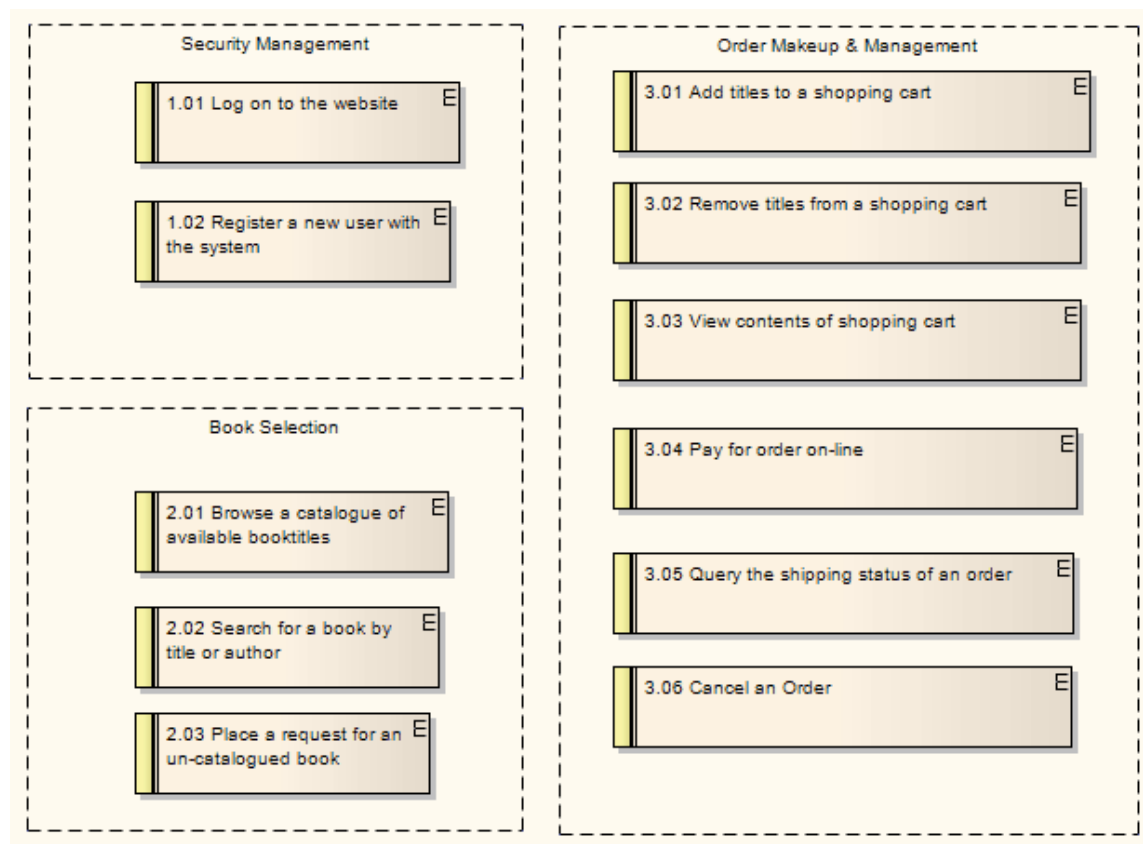
Example 2
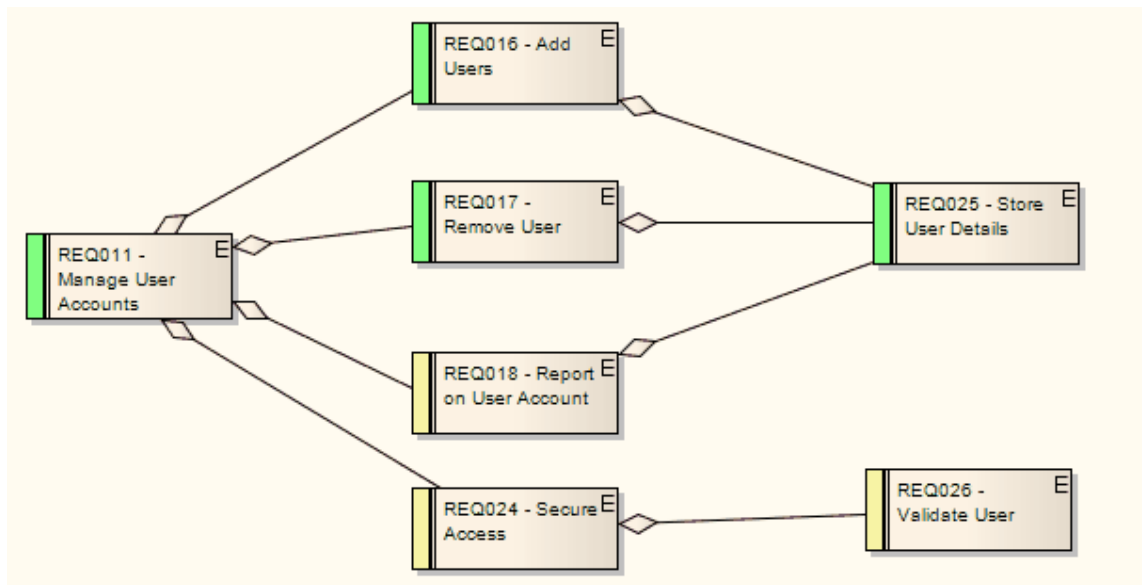
Notes

- Change and Defect (Issue) elements resemble Requirement elements and can be coded and managed in the same way

Feature
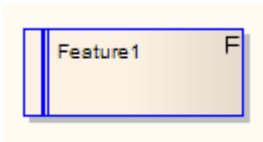
Description

A Feature is a small, granular function or characteristic expressed in client-valued terms as a satisfaction of a requirement; for example: 'context-sensitive Help', or 'ability to reverse-engineer VB.Net'.

Features are the primary requirements-gathering artifact of the Feature-Driven Design (FDD) methodology. They define the product feature that satisfies what a Requirement element has formalized as a contractual, testable, expected deliverable (for example: requirement - 'every element must provide context-sensitive Help'; feature - 'every element provides context-sensitive Help'). One Feature might realize one or more Requirements, and one Requirement might be realized by more than one Feature.

Features also have relationships with Use Cases. A Use Case defines the interaction a user has with the system in order to satisfy one or more Requirements. The Feature identifies the facility that provides the means for that interaction.

Feature elements are non-UML and are not related to the UML elements of the same name, which are either

BehavioralFeatures (operations, or methods) or StructuralFeatures (Ports, Parts and attributes).

Feature elements are available from the 'Requirements' page of the Toolbox.

Feature elements can be displayed with or without an identifying 'F' in the top right corner of the element. To toggle the display of this letter, select or deselect the 'Show stereotype icon for requirements' checkbox on the 'Preferences' dialog, 'Objects' page.

Toolbox icon
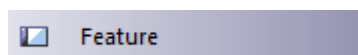
Internal Requirements

In Enterprise Architect, internal requirements are requirements that are specific to one element. For example, an internal requirement to enable the user to log in to the system could be defined for the Login Use Case. This is a requirement of the Use Case - an action it is responsible for carrying out - and it applies only to this Use Case.

Internal requirements form the functional requirements of the system to be built. The meaning of the requirement can vary depending on which element is the host; for example, a business process requirement might mean something different to a Use Case requirement, which again might mean something different to a Class requirement.

Internal Requirements are defined on the 'Requirements' page of the Responsibilities window. The significant parameters (or, in Requirement Management terms, attributes) are the Type, Status, Difficulty and Priority. Whilst you can provide a detailed description of the requirement in the 'Notes' field, there is more scope in the name ('Requirement' field) to define the nature of the responsibility. An additional field, 'Stability', indicates the probability of the requirement changing; high stability means a low probability of change.

The example Use Case also has connections to two external Requirement elements, which are system functions that the Use Case implements either in full or in part. You can convert an internal requirement into an external

Requirement element.

You can also create internal requirements for an element using an instance of the Responsibilities window within the element 'Properties' dialog. A requirement created in the window displays in the 'Properties' dialog, and vice versa.

Make Internal Requirement External

Elements in Enterprise Architect have internal requirements (what they must do or accomplish). These can overlap or duplicate more formal requirements that the system in general must meet, so you might decide to make a single element's internal requirement into an external Requirement element (where the requirement can perhaps be implemented by multiple elements). You can make this conversion in one operation, using the 'Move External' function.

Access

On a diagram or in the Browser window, select an element and:

Ribbon	Design > Element > Editors > Requirements
Keyboard Shortcuts	Alt+Enter > Requirements Shift+Alt+R
Other	Double-click on element > Requirements

Change an element's internal requirement into an external Requirement element

Step	Action
1	('Properties' dialog) In the 'Requirements' list, right-click on the internal requirement to change to an external requirement OR (Responsibilities window, 'Requirements' tab) Click on the drop-down arrow at the end of the 'Requirement' field and, in the list, right-click on the name of the internal requirement to change to an external requirement A short context menu displays.
2	Click the on 'Move External' option. The 'Find Package' dialog displays.
3	Locate and click on the Package to place the new Requirement element in.
4	Click on the OK button. A new Requirement element is created in the target Package, with a Realization connector from the current element to the Requirement.

Notes

- When an internal requirement is made into an external Requirement element, the 'Stability' field and its value for the internal requirement are translated into the Stability Tagged Value in the external Requirement

Create Requirements

Requirement Analysts tend to work in a variety of ways, some preferring to work with lists of elements as they would in a spreadsheet, or with textual representations as in a word processor document, or visually as in a drawing tool. Enterprise Architect embraces this variation and provides functionality for Requirement Analysts to work using their preferred method for creating and editing requirements. This can be in diagrams, in lists such as the Diagram List and Package List, and in the Specification Manager. The Specification Manager is a powerful and flexible textual tool that simulates working in a word processor but that allows you to edit names, descriptions and properties of a requirement in a single interface.

Enterprise Architect also assists with creating Requirement elements by allowing them to be dragged into the model from external text documents or by importing them from spreadsheets and other documents, and it can integrate with large requirements management tools such as IBM Rational (formerly Telelogic) DOORS.

Ways to create Requirement elements

Within Enterprise Architect you can create external Requirement elements in a number of ways, such as:

- Typing or copy/pasting a text description into the body of the Specification Manager

- Dragging a Requirement icon from the Diagram Toolbox into a specific diagram

- Generating an element within a specific Package in the Browser window

- Dragging text from a text file onto a diagram, to generate a requirement based on that text

- Importing requirements from a spreadsheet application such as Excel, via CSV

- Creating Requirement elements on the Package Browser or Diagram List for the selected Package or diagram

- Converting an internal element requirement (responsibility) into an external Requirement element, in a selected target Package

- Importing requirements from another requirements management tool, such as IBM Rational (formerly Telelogic) DOORS (in this case via the Sparx Systems MDG Link For DOORS integration tool)

All methods that add a Requirement to a diagram or window also add the Requirement to the diagram's parent Package in the Browser window.

Create Requirement elements from text

This procedure converts a text section heading into an element name and the section text into the element's Notes text. You can use this procedure to generate elements of a range of types; however, it is particularly useful for

generating Requirements from a requirements specification document.

Step	Action
1	Open a Requirements diagram in the Diagram View.
2	Open the document file containing the text you want to generate Requirement elements from (this can be opened in any common text editing tool).
3	Highlight the required heading and associated text and drag them from the text file into the diagram. The 'Toolbox Shortcut' menu displays.
4	Navigate through the menus and select the required element type (in this case, click on Common and Requirement).
5	Enterprise Architect creates a (Requirement) element in the diagram, and displays the 'Properties' dialog with the section heading in the 'Name' (or equivalent) field and the text in the 'Notes' field; the element is also added to the diagram's parent Package.

Notes

- The Requirement element name can be simply descriptive text, with or without a manually-typed reference number; however, as requirements often have to have a unique reference for external checking, you can use the Enterprise Architect auto-numbering facility to automatically apply a numbering system with or without prefixes and suffixes - set the element type to Requirement

- Requirement elements can be displayed on a diagram with or without an identifying 'E' in the top right corner; to toggle display of this letter, select or deselect the 'Show stereotype icon for requirements' checkbox on the 'Preferences' dialog, 'Objects' page

- Requirement elements can be color coded on a diagram to indicate their status

Requirement Properties

Requirement properties differ slightly from the properties of other elements; they include information related to the difficulty and priority of the Requirement. The 'Notes' field is also important, as it describes precisely what requirement the element represents. Requirement naming requires careful consideration and could reflect either a categorical naming convention, or simply a loose English description of the Requirement.

Access

On a diagram or in the Browser window, select a Requirement element and:

Ribbon	Design > Element > Editors > Properties Dialog
Keyboard Shortcuts	Alt+Enter
Other	Double-click on Requirement element

Fields

Field	Action
Name	The name of this Requirement, which could include numbering, plain English text or some other formal specification. If you have set autonaming and autonumbering, the field contains the auto-counter text.
Notes	The description of this Requirement.
Stereotype	Either type in the name of a stereotype to be assigned to the Requirement, or click on the ⬚ button and use the Stereotype Selector to locate and select the stereotypes to assign.
Alias	An alias to be used for this Requirement. If you have set alias autonaming and autonumbering, this field contains the auto-counter text.
Status	The current status of this Requirement.
Version	The version of this Requirement.
Phase	The phase of this Requirement.

Language	Click on the drop-down arrow and select the appropriate programming language, or the <none> option.
Filename	Click on the ⊡ button and select the appropriate file location for the Requirement.
Difficulty	An estimate of the difficulty in meeting this Requirement; select from: • Low • Medium • High
Priority	The relative importance of meeting this Requirement compared to other Requirements; select from: • Low • Medium • High
Package	The name of the Package that contains the Requirement element.
Complexity	Click on the drop-down arrow and select the appropriate development complexity (used for project estimation). You

	normally select from three levels: • Easy • Medium • Difficult You can change your user defaults to add 'Extreme' and 'Unknown' to this list of options.
Created	Read-only field specifying when this Requirement was first created.
Modified	Read-only field specifying when this Requirement was last changed.
Key Words	A set of words that could be used to index or define the subject of this Requirement.
GUID	The globally unique identifier (GUID) of the Requirement element.
Author	The author of this Requirement.

Notes

• In Requirement Management tools and texts, the

characteristics of a requirement are commonly called attributes; however, in UML the term attribute refers to a different type of feature, and the Requirement characteristics are defined as properties - in this Enterprise Architect documentation, the term properties is used

· In a project, it might be necessary to define more information in a Requirement than is provided by the standard properties; for more information on extending the Requirement properties, see the *Extend Requirement Properties* Help topic

Color Code External Requirements

External Requirement elements can be color coded on a diagram, to provide quick visual cues indicating the status of a requirement. The color code requirements use these default conventions:

- Yellow for 'Proposed'
- Blue for 'Approved'
- Green for 'Validated'
- Orange for 'Mandatory'
- Black for 'Implemented'
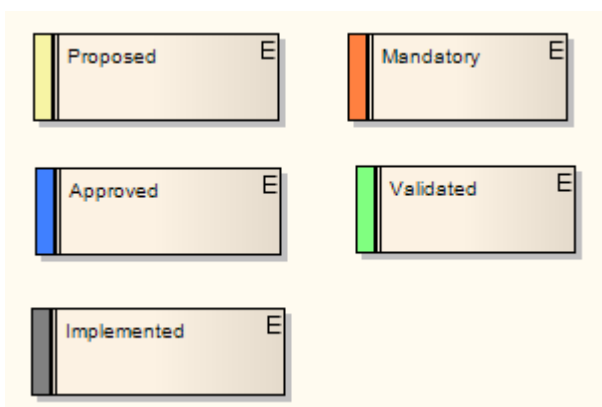
You can change these colors, and add or remove status types, using the 'Status Types' dialog.

Access

Ribbon	Start > Desktop > Preferences > Preferences > Objects > Show status colors on diagrams

Enable color coded external requirements

Step	Action
1	Select the 'Show status colors on diagrams' checkbox to enable the status of external requirements to be represented by color coding.

Extend Requirement Properties

A project might apply further properties to a requirement, such as cost, lateness penalty or risk to the business if not met. You can add these properties to specific Requirement elements, or configure them to be automatically available in all Requirement elements on creation, using Tagged Values. (These are sometimes referred to as User-defined attributes.)

Extended element properties are not visible unless you open the 'Tags' tab of the Properties window for the element. Alternatively, you can display the additional properties on the element image on its diagrams.

Add Tagged Values to Existing Requirements

To add a property to a single Requirement as a Tagged Value, simply click on the Requirement, display the 'Tags' tab of the Properties window, and enter the name of the property as the tag name and the value of the property as the tag value.

It is likely that any property you add to one Requirement would also apply to others. You might therefore use a predefined Tagged Value Type to identify your Requirement property, so that you can select it whenever required. The predefined Tagged Value Type also enables you to define specific values for the Tagged Value.

If the appropriate predefined Tagged Value Type does not

exist, a Technology Developer can create it to add to the structured tags, reference tags, or customized tags collections.

Configure Requirements to be Created with Extended Properties

If it is necessary to create all Requirements with the same extended set of properties, you can create a Requirement Template diagram and either create a special Requirement that defines those properties (as Tagged Values), or drag an existing Requirement with those properties onto the diagram. You then set the Requirement Template diagram as the template for all new Requirement elements, so that those new Requirements automatically have all of the properties you want.

However, this then excludes other Requirement element formats, including the standard Requirement format. If you want to use another Requirement format, you have to replace or cancel the current Template. Alternatively, you can create a Profile.

A Profile also defines exactly what a new Requirement element should contain, and how it should display in diagrams. However, a Profile is a collection of alternative element definitions, so it does not override the default Requirement format, nor does it prevent you from defining several different types of Requirement element. You can

therefore have separate and parallel definitions of elements for business requirements, system requirements, project requirements, or any other category of requirement you decide to work with.

For information on importing and using existing Profile files, see the *Using UML Profiles* Help topic. For information on creating new Profiles, see the *Developing Profiles* Help topic.

Display Tagged Values On Diagrams

If you have extended the properties of a Requirement, you might want to make those properties visible in the Requirement elements in your diagrams, by switching on display of the element tags compartment.

You can do this in one of three ways:

- To display the tags compartment on all elements on a diagram, double-click on the diagram background and select the 'Elements' tab of the diagram's 'Properties' dialog; select the 'Tags' checkbox and click on the OK button

- Alternatively, to show the tags compartment on all elements in the diagram, click on the diagram background, press Ctrl+2 to display the Properties window (if it is not already visible), and then click on the 'Compartments' tab of the window; select the 'Tags' checkbox

- To display the tags compartment on a specific element on a diagram, regardless of the compartment setting for the diagram, right-click on the element and select the 'Compartment Visibility' option; select the 'Tags' checkbox in the 'Show Element Compartments' panel of the 'Compartment Visibility' dialog, and click on the OK button

The Tagged Values are then displayed in the Requirement element on the diagram.

REQ016 - Add Users

tags
Cost = $25000
Lateness penalty = $300 per day
Loading Factor = 4
Risk = High

Connect Requirements

Aspects

Aspect	Detail
Abstract	A Requirement element can be connected to other Requirements, most commonly using Aggregate relationships to form a hierarchy of requirements.
	Requirements are also connected to other types of element, most commonly Use Cases, by Realize or Implements relationships.
	These relationships are very important, both in identifying how the Requirements are organized and used in the model, and in tracing the development from the Requirements throughout the model. Both of these tasks are very simple in Enterprise Architect, because once a connector on a Requirement exists, Enterprise Architect automatically lists the Requirement in the:
	• Requirements Traceability window (an important tool in examining the role of Requirements in the model)

	• Specification Manager • 'Requirements' tab of the target element 'Properties' dialog • 'Links' tab of the Requirement element 'Properties' dialog • Responsibilities window • Relationships Window • Dependency and Implementation reports • Standard document output The connector itself is also listed in the 'Links' tab of the target element 'Properties' dialog, and in the Relationship Matrix. There are, therefore, many ways to locate, view and track Requirement relationships.
Connect On Diagram	Relationships can be created on a diagram by clicking on the appropriate connector icon from the Requirement and Common pages of the Toolbox, clicking on the source (originating) element, and dragging to the target element. If you are connecting elements in different Packages, you can drag elements from the Browser window onto a common diagram and set up the relationships there.

Quick Generation Of Realize Connector	You can quickly generate a Realize connector by dragging an existing Requirement element from the Browser window into a diagram, over the element that implements the Requirement (usually a Use Case). Enterprise Architect interprets this as a request to create the Realize connector and does so automatically. The Requirement element is not added to the diagram. However, if you subsequently drag the Requirement onto the diagram the connector is already in place.
Connect Off Diagram	You can also connect a Requirement element to other elements without necessarily having the elements on the same diagram, or having a diagram open. Use the Relationship Matrix to create relationships for requirements; this is a convenient way of quickly building up complex relationships and hierarchies.

Import Requirements and Hierarchies in CSV

You can import Requirements from a spreadsheet application in CSV format. Before doing this you must create a CSV import file specification that:

- In the 'Default Types' field has the value 'Requirement Package' to import requirements and a Package structure to contain them

- Has the 'Preserve Hierarchy' checkbox selected

- Identifies the data fields on the spreadsheet that are to be translated into Enterprise Architect, in the order in which they are plotted across the spreadsheet

- Is to operate on a spreadsheet containing the 'CSV_KEY' and 'CSV_PARENT_KEY' fields (which, if not generated by a CSV export from Enterprise Architect, you have added and populated yourself)

This enables you to import the individual and grouped requirements from the spreadsheet into Enterprise Architect, and to reconstruct the hierarchies of Requirements in the target Package in the Browser window.

Manage Requirements

Requirements Management is the process of maintaining requirements and ensuring they are traced, prioritized, assigned to phases, iterations or sprints in a development life cycle, and also ensuring those changes are communicated effectively to stakeholders. Enterprise Architect has built-in functionality that allows you to assign properties to each requirement for management purposes, such as status, priority, phase, difficulty and detailed notes. This model-based approach to requirements management allows the requirements to be traced to up-process elements such as stakeholders and goals and to down-process elements such as Use Cases and application components. Requirements Managers can work in a variety of ways and Enterprise Architect allows them the flexibility of viewing requirements as visual elements in diagrams or in lists and textual representations. The Specification Manager and list views allow the requirements metadata to be visualized and edited in a single interface. There is also a powerful chart and graph facility where the requirements metadata such as status and priority can be displayed in compelling charts and graphs. For example, a pie chart could be created showing the percentage of elements with particular statuses.

Example
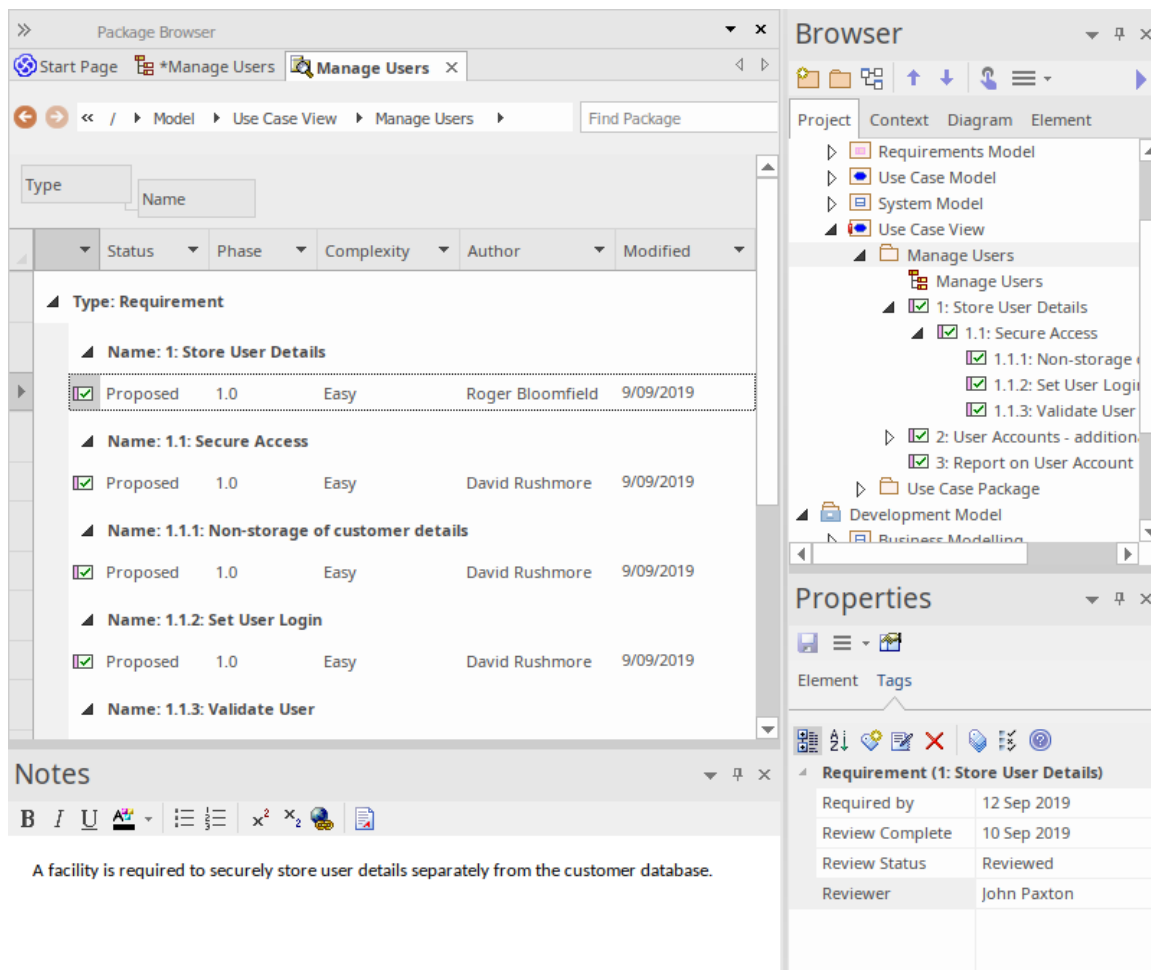
This display shows the position of the Store User Details Requirement element in the model, and how it relates to other Requirements (Browser window); the default characteristics of the Requirement (Package Browser) and the extended characteristics ('Tags' tab of the Properties window), and a detailed description of the Requirement (Notes window). You can configure some of these windows to display more information, and/or use other windows and facilities.

View Requirements

Use these windows and facilities to: locate and list Requirement elements in the model; add, move and delete the elements; display and edit the properties and characteristics of individual elements; and generate reports on Packages or specific elements.

- Project tab of the Browser window - shows the content and structure of your model

- Specification Manager - shows Requirements (and other element types) in a simple text format, and helps you to create and manage these elements

- Diagram List - lists the elements in a diagram, filtered and sorted according to the settings you define; shows all or selected default properties of each element

- Package Browser - lists the elements in a Package, filtered and sorted according to the settings you define; shows all or selected default properties of each element

- (Requirements) Diagram - shows the arrangement of a group of Requirements, and can show whether the elements are in the same Package or different Packages

- Model Search - enables you to locate Requirements in general in the model, or specific Requirement elements, according to the search criteria you use

- Model Views - enables you to maintain links to commonly-used elements, and to rapidly show developments and changes in (Requirement) Package

contents through either reports or slide shows of selected diagrams

- Properties - shows every standard property of a selected element, whether updated by the user or maintained automatically by the system

- Tagged Values - shows extended properties of a selected Requirement element

- Element tab of the Browser window - shows every added-on property, such as attributes, operations, Tagged Values and constraints

- Notes - displays the detailed description of a requirement, and any other additional information recorded on the requirement

Trace Use of Requirements

Having investigated the representation of Requirements in your model, you could review either how they have been used to direct development through the model, or how a particular development was initiated. The windows and facilities you might use to follow development from Requirements are briefly described here; detailed information is also available in the Traceability topics.

The significant feature in tracing Requirements and development is the connectors between the elements.

Facilities

Facility	Detail
Relationships Window	Using the Relationships window, you can quickly identify every relationship of which a selected Requirement element is a member and the partner element in that relationship, whether or not the relationship is visible in the current diagram. If the partner element is not in the diagram, you have the option of adding it.
Traceability	The Traceability window is a very useful

Window	tool in showing chains of relationships that include the selected element. The window can show, for example, that:
	• Requirement A is realized by a Use Case X, and
	• Use Case X also realizes Requirement B, and
	• Requirement B in turn is also realized by Use Case Y
	You can control the type and extent of these relationship chains. As the system checks the connectors and partner elements of every relationship within the limits you impose, if you specify broader limits the system can take some time to produce the final results.
Relationship Matrix	The Relationship Matrix is a significant tool in mapping the relationships between the Requirements elements in a Package and other elements in either that Package or a different Package. Where a relationship is missing, you can add it; if an existing relationship is misplaced, you can delete it.
Requirements tab,	On the 'Properties' dialog for elements other than Requirements - particularly

Properties dialog	Use Cases - the 'Requirements' tab shows all internal requirements of the element (and, where an internal requirement has been converted to an external Requirement element, that element).
Responsibilities Window	The Responsibilities window - as for the 'Properties' dialog - shows the internal requirements of the selected element, and the scenarios and constraints under which the requirements are being realized.
Validation	It is useful to review the way that you have modeled your requirements, to check that they are correctly set up and connected to other elements. The Validation facility has a number of configuration options for validating various aspects of model development, as well as an option for specifically validating Requirements Management. This can reveal, for example, which of your Requirement elements do not yet have a Realization connector.

Manage Requirement Changes

Because requirements are statements of what a system or process must do or provide, they have a great impact on the modeling and development of the system. A new requirement might initiate an extensive program of work, and changes to or removal of that requirement can therefore have a major effect on the model. Issues concerning requirements, and changes to Requirement elements, must both be carefully managed.

The first steps in managing changes to requirements would be to raise specific Issue and Change request items against the Requirement element. You could monitor the appearance of these items using the filtered searches of the Model Views. You might then review the Requirement properties and/or its relationship hierarchies. During model development, you might capture periodic Baselines and use these to review the changes and, if necessary, roll them back to a previous point. You might also use the Auditing facility to monitor changes as they are made, and to ensure that no unauthorized or potentially risky changes are being made in the model.

Facilities

Facility	Detail

Changes and Issues	A change is, very broadly, an item defining an addition or alteration to a requirement. An issue identifies either a failure to meet a requirement, or a risk in meeting the requirement.
	Changes and issues can arise in development at a number of levels, being raised for problems that apply system-wide down to within a specific element. There are two mechanisms that can be used to identify a change or issue, and the work required to resolve it:
	• Change and Issue (or Defect) elements - structured comments that identify a problem at system-level, although they can also be attached to a specific element from which a problem arises. Both types of element resemble the Requirement element, and can be linked to one or more other elements that have to be reviewed, with relationships such as Association, Dependency and Realize. The two types of element can also form hierarchies or groups, where complex problems arise
	• Maintenance items raised against a specific element, and recorded for that element in a Maintenance window.

	Maintenance items enable the distinction between Defects (a failure to meet a requirement) and Issues (a risk factor that might affect satisfying the requirement). They also include Tasks, which record work items associated with the element Maintenance items are very specific, but if there is a possibility of an item having a wider impact on other elements or the system in general, you can translate the item into a Change or Issue element, or any other type of element that best identifies the problem and its solution.
Model Views	Model Views are very useful for trapping changes and issues in the model, especially on Requirements. You can set up searches to identify the appearance of new Change or Issue elements, or to detect changes in the properties of the Requirement elements themselves.
Baselines	A Baseline is a snapshot of a Package or a model branch at a particular point in time, which you determine. You can use the Baseline as a distribution mechanism for changes to the model, but the main use is to enable you to compare the

	current model with a previous stage, and detect what changes have been made since the Baseline was captured.

If you do not want a change to remain in the model, you can roll the affected elements back to the state they had in the Baseline. Therefore, if you maintain your requirements in a specific Package or branch, you can capture Baselines of the Package and ensure that changes conform to your change management process or, if not, can be reversed. |
| Auditing | The Auditing facility enables you to capture any changes made to your model within the selection criteria that you define. You can, for example, configure the Auditing facility to specifically record changes to Requirement elements.

As auditing is continuously monitoring, you can detect changes as they are made, and verify that they are acceptable. You can also store the log of changes, and review it later on.

Note that you cannot reverse the changes automatically, as you can with Baselines. You might therefore use Auditing to identify changes to investigate more fully and - if necessary - reverse in a Baseline |

	comparison.

Report on Requirements

Enterprise Architect provides two report generation facilities that help you to output document reports and web reports on your model structure and components.

The document reporting facility is especially comprehensive, and contains a number of features that provide particular support to Requirements Management:

- A requirements report template that extracts details of external requirements in the model; you can copy and tailor this template for your particular requirements

- Options in the Specification Manager, Diagram List, Package Browser and Model Search to generate reports on selected (Requirement) items from the collected information

- The Implementation Report, which lists for a selected Package the elements that require implementers, together with any source elements in Realize (Implements) relationships with those elements

- The Dependency Report, which lists for a selected Package any elements that are dependent on another element for their specification; for example, a Use Case derives its specification from the Requirement that it realizes

Decision Tree

A Decision Tree uses a visual notation to represent a series of decisions and possible outcomes. It can be used in either a descriptive or predictive manner to visualize outcomes and decision points.

Creating a Decision Tree

1. Create a new Package called 'Decision Tree', followed by a Decision Tree diagram called 'Key Decisions'.
2. Select the Pattern 'Decision Tree - Object' and place it on the Diagram View.
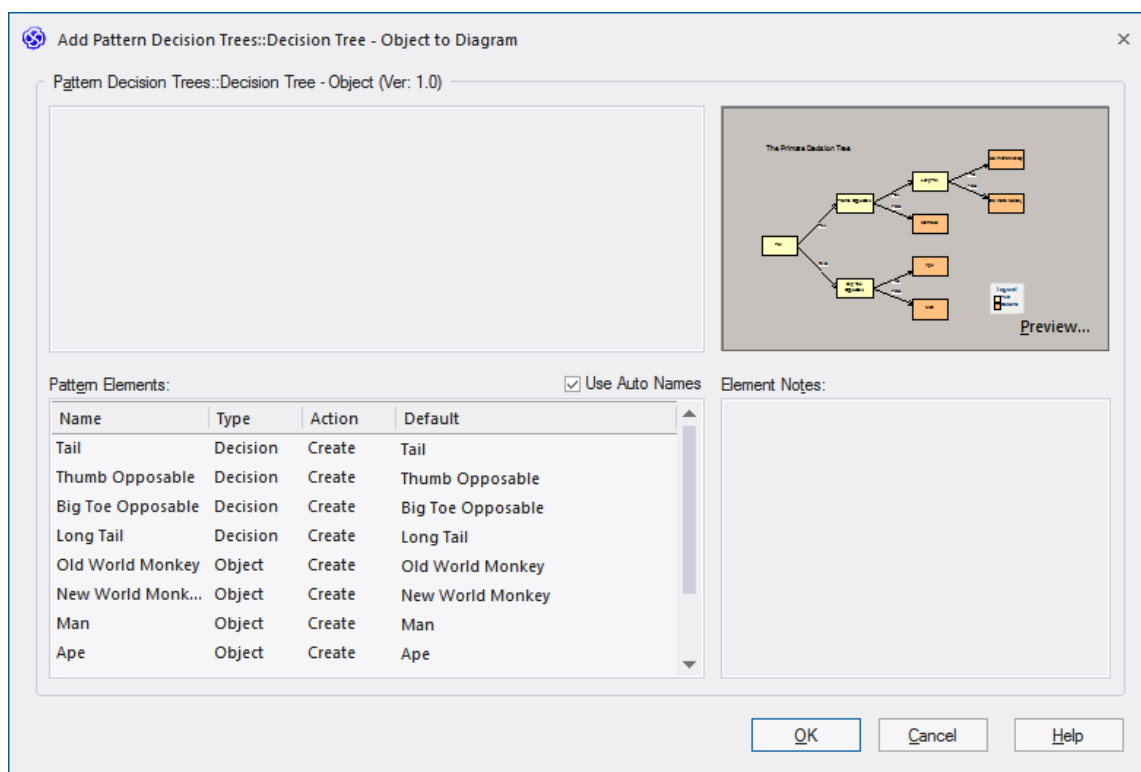
Modify the 'Primate Decision Tree' template created by the
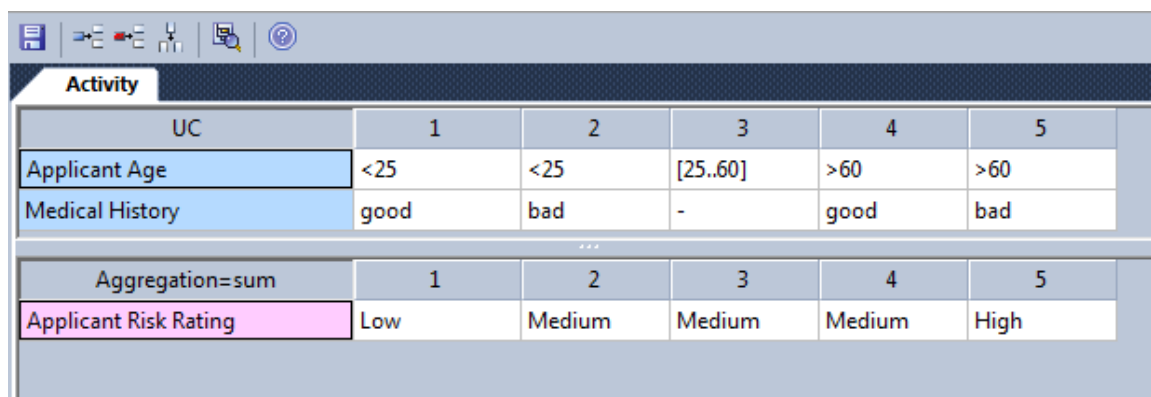
Pattern to create a Decision Tree related to the interview process for staff at a restaurant.
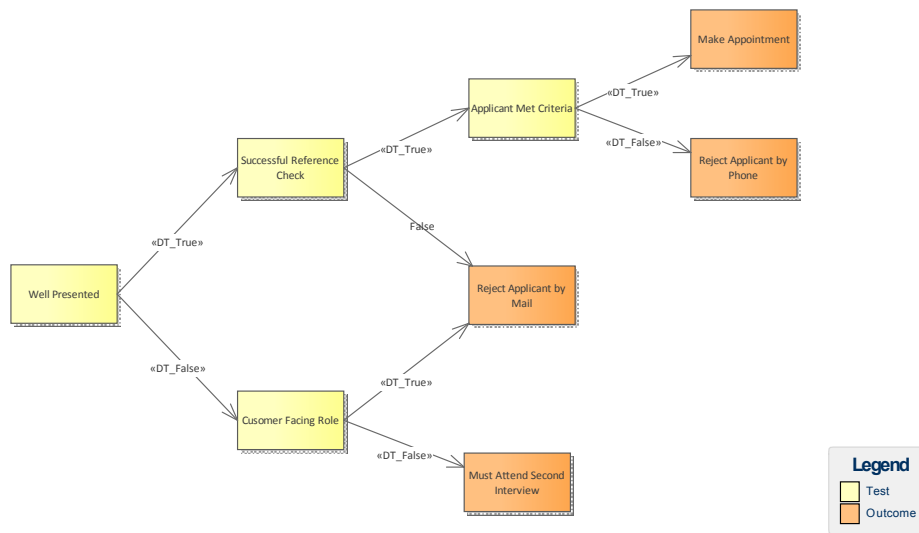
Decision Models

Enterprise Architect provides the Decision Tables facility as a simple method of applying the Decision Model and Notation (DMN) standard to model how a decision is made, and generate code for the decisions.

This example illustrates how you could model a decision on an insurance Risk Rating based on the applicant's age and medical history.

UC	1	2	3	4	5
Applicant Age	<25	<25	[25..60]	>60	>60
Medical History	good	bad	-	good	bad

Aggregation=sum	1	2	3	4	5
Applicant Risk Rating	Low	Medium	Medium	Medium	High

On the diagram containing the Activity or Action element that owns the table, you can replace the element shape with the Decision Table itself, showing the rules as either columns or rows. You specify the table format, or whether to redisplay the element shape, using options on the element context menu.

Enterprise Architect also supports a more extensive and complete implementation of the Decision Model and Notation (DMN) standard; see the *DMN Modeling and Simulation* section of the Help.

Access

Ribbon	Design > Impact > Decisions > Manage Decisions
Context Menu	On a diagram: • Right-click on Activity element | Simple Decision Table | Decision Table or • Right-click on Action element | Simple Decision Table | Decision Table

Notes

- The Decision Tables facility is available in the Corporate, Unified and Ultimate editions of Enterprise Architect

Decision Table Editor

When you select to create a new Decision Table for an Activity element, or edit an existing one, the name of the Activity element displays in the tab under the Decision Table toolbar.

Access

Ribbon	Design > Impact > Decisions > Manage Decisions
Context Menu	On a diagram: Right-click on Activity element | Simple Decision Table | Decision Table or Right-click on Action element | Simple Decision Table | Decision Table

Complete the Decision Table

Task	Action
Enter a	You can enter a condition into the table in

condition	a number of ways: • Type the text of the condition into the first available row, overtyping the *Add Condition* text • Drag onto the 'Add Condition' cell the appropriate behavioral object from the Browser window, such as a model element, Activity Parameter, operation or attribute, or • Click on the 'Add Condition' cell and press Ctrl+Space; Intelli-sense is invoked to provide a list of possible objects that you can select from to complete the condition
Add another condition	You can add further condition rows to the list in any of these ways: • Click on the ⊕ icon underneath the list of conditions • Right-click anywhere on the condition panel and select the 'Add Condition' option, or • Click on the ⊞ (Add Condition) icon in the Decision Table toolbar Each condition that you add is bound to the previous conditions, so that they are evaluated in combination.

List allowable values for the condition	(Optional) You can provide a comma-separated value (CSV) list of the values that you could assign to a condition. If you do this, those values are offered as suggestions to enter in the value fields along the same row. The allowable values column is hidden by default. To display it, click on the icon in the Decision Table toolbar. In the allowable values cell against each condition, type in all the possible values separated by commas. These values can include operators such as >, <, = and ... . Attributes dragged from the Browser window and classified by an Enumeration will complete the cell automatically and make the field read only.
Set condition values	Each of the conditions you set have defined values that act as a decision point. For example, a person under the age of 18 years old cannot hire a car, so the values of 'Yes' and 'No' are decision points for the condition '18 or over'. In each of the columns 1, 2, 3 and so on, provide a condition value by either: • Typing the value into the cell (including operators such as >, <, = and ...) or

	• Right-clicking on the cell and selecting the 'Allowable Values | <value>' option (as set in the allowable values cell)
Add further condition value columns	To add further value columns to the table, either: • Right-click anywhere on the condition panel and select the 'Add Column' option, or • Click on the (Add Column) icon in the Decision Table toolbar These options add the column to both the 'Condition' and 'Conclusion' panels.
Set the Policy and Completeness	The 'Decision Hit Policy' defines if and how the conditions are combined in making a decision. 'Completeness' identifies whether or not the decision set is complete. To set these two flags, right-click on the 'Condition' header (top left corner of the 'Condition' panel) and select the options: • 'Hit Policy | <value>' and • 'Completeness | Complete' or 'Incomplete' If you do not want to display the completeness, select the 'Completeness | Clear' option.

Delete Column or Condition	To remove a condition that no longer applies, or a column of values that are no longer tested, right-click on the appropriate column or row and select the options: • Delete Condition or • Delete Column A prompt displays to confirm the deletion. You cannot delete the title column or condition number row, nor can you delete the allowable values column.
Enter a Conclusion	You can enter a conclusion into the table in a number of ways: • Type the text of the conclusion into the first available row, over typing the *Add Conclusion* text • Drag onto the Add Conclusion cell the appropriate behavioral object from the Browser window such as a model element, Activity Parameter, Operation or Attribute
Add Further Conclusions	You can add further conclusion rows to the list in any of these ways: • Click on the ⊞ icon underneath the list of conclusions

	• Right-click anywhere on the conclusion panel and select the 'Add Conclusion' option, or • Click on the ■■ (Add Conclusion) icon in the Decision Table toolbar Each conclusion that you add is bound to the previous conclusions, so that the final decision takes them all into account.
List allowable values for the conclusion	(Optional) You can provide a comma-separated value (CSV) list of the values that you could assign to a conclusion. If you do this, those values are offered as suggestions to enter in the value fields along the same row. The allowable values column is hidden by default. To display it, click on the ■■ icon in the Decision Table toolbar. In the allowable values cell against each conclusion, type in all the possible values separated by commas. These values can include operators such as >, <, = and ... . Attributes dragged from the Browser window and classified by an Enumeration will complete the cell automatically and make the field read only.
Set Conclusion	Each of the conclusions you set has values that define the outcome of the

values	conditional values in the same column, when the conditions are met. In each of the columns 1, 2, 3 and so on, provide a conclusion value by either: • Typing the value into the cell (including operators such as >, <, = and ...) or • Right-clicking on the cell and selecting the 'Allowable Values | <value>' option (as set in the allowable values cell)
Add Further Conclusion Columns	To add further value columns to the table, either: • Right-click anywhere on the conclusion panel and select the 'Add Column' option, or • Click on the ⚏ (Add Column) icon in the Decision Table toolbar These options add the column to both 'Condition' and 'Conclusion' panels.
Set theTable Aggregation Values	The 'Table Aggregation' value indicates how the conclusion values are to be combined to form the decision. Set the value by right-clicking on the 'Conclusion' column heading and selecting the'Aggregation | <value>' option.

	If you do not want to set the Table Aggregation, select the 'Aggregation | Clear' option.
Delete Column or Conclusion	To remove a conclusion that is no longer valid, or a column of result values that are no longer produced, right-click on the appropriate column or row and select the options: • Delete Conclusion or • Delete Column A prompt displays to confirm the deletion. You cannot delete the title column or conclusion number row, nor can you delete the allowable values column.
Check Object properties and location	If you have created a condition or conclusion by dragging an object from the Browser window, you can right-click on its row and: • Display the 'Properties' dialog for the object, by selecting the 'Properties' menu option • Locate the object in the model by selecting the 'Find in Project Browser' option
Save your	At regular intervals, and before you exit

| changes | from the table, click on the Save icon in the Decision Table toolbar. |

Notes

- The Decision Tables facility is available in the Unified and Ultimate editions of Enterprise Architect

Code Generation from Decision Models

It is possible to generate the method code of a Class from an Activity element, based on the logic defined in its Decision Table, by placing the Activity as a child of the Class element from which the code is generated.

Set up Class and generate code

Step	Action
1	Create an Activity with a Decision Table.
2	Make the Activity element a child of a Class element.
3	Select the Class and press F11 to open the 'Generate Code' dialog, and generate the code.
4	Press F12 to view the generated source code.

Decision Table code templates

Code Generation from a Decision Table automatically applies these EASL code generation templates:

- Behavior Body
- Decision Table
- Decision Logic
- Decision Condition
- Decision Action

Notes

- Currently C++ is the only language implemented in the EASL templates for Decision Table code generation
- You can use ActivityParameters to define method parameters
- You can add Attributes to the Activity element to define local variables
- The Decision Tables facility is available in the Unified and Ultimate editions of Enterprise Architect

Modeling Business Rules

In any business action or process, the start, progress and end result are usually determined by reference to a set of rules. These rules can be very simple, such as 'the client must present documentary evidence of being at least 18 years old', or very complex, such as the actuarial rules that determine what a tailored insurance policy will and will not cover.

Whether the rules of your business are simple or complex, there are two sets of considerations to take into account:

- How to manage the rules - How are they initially identified? Where are they held? Are the rules easily maintained and updated? How are they refined and tested?

- How to use the rules - How easy is it to identify which rules apply in a specific context? How easily can any specific rule be recognized and applied? How are the rules executed in the process - can they be integrated with the process? Can execution of the rules be automated in the process?

Both sets of considerations can be easily managed by modeling your business processes in Enterprise Architect, and using the Business Rule Model facility. Business Rule modeling captures the rules that govern a business, and their relationships with the entities and specific tasks within the organization or system.

Managing Rules

Broadly, modeling your business processes can clarify:

- Your business requirements (from which many business rules are ultimately derived)
- The Use Cases - and the scenarios in each Use Case - to satisfy those requirements, and
- The exact processes, stages, objects, actions and data structures that support those Use Cases, represented by Classes

This process will also clarify which of your current business rules are applicable to which points in each process, and what refinements or new business rules are required. You can then map your business rules to existing or new Classes, using two specific Business Rules models; the:

- Business Domain model, in which you group the business objects (represented by Classes) involved in a process or application, and develop a Rule Flow that defines the tasks (as Rule Task elements) associated with the process as a whole or specific objects in the process
- Business Rules model, in which you create a specific Business Rule element for each business rule and associate it with the Rule Task to which the rule applies

When you have defined all the tasks, their sequences, and the rules that apply to each one, you can compose the rules per task to define the values and conditions of the rules and how they take effect in the task. You can then validate the rules for the task to ensure that they are logical.

A valuable resource that you have created in this process is

a database of business rules associated directly with the tasks and procedures they apply to, easily explored (according to the naming and/or numbering convention you have used) with the Model Search and other navigation and display facilities, and documented through the document or web reporting facilities. You can also record further information on each rule using internal or external notes, Tagged Values and Linked Documents.

Using Business Rules

Having set up the business rules database, your users can access the models or their documentation as a reference. As explained previously, the context of any given rule, or the rules applicable to a context, can be quickly established using the search, navigation or Traceability facilities.

However, you can use Enterprise Architect to model and create applications and user interfaces that can apply the business rules you have defined, and a further facility of Business Rule modeling is to generate the behavioral code for the rules in a specific task. You can merge this into your code to prompt for or even automate the correct use of the business rules in performing a task.

Advantages of modeling Business Rules

Whether you create a database of rules, or applications that apply the rules, you have a modular solution to a business process requirement. This provides an advantage in localization. Business Rules can vary between locations; for example, car hire operates in roughly the same way in most countries, but the legal driving age differs between the countries, as do the models of car available for hire. You can easily create different localized rule modules and switch the appropriate one for the current location into the common model.

Notes

- Business Rule Modeling is available in the Unified edition and the Ultimate edition of Enterprise Architect

Develop a Business Rules Model

In modeling Business Rules, you first develop the model structure to represent the rules in the context of their use, and then effectively compile (or compose) the rules to make them operational within that context. From the compiled rules you can either create a spreadsheet for reference or generate behavioral code for applications that apply the rules, or both.

Process Summary

1. Use the Model Wizard to generate a Business Rules Model in which to define the business rules.

2. In the generated Business Rules diagram, begin to identify the business rules as Business Rule elements, each element representing a specific business rule.

3. In the generated Business Domain Model diagram, edit the existing Classes - and create others - to represent the Business Objects in the business domain or process; these provide the business vocabulary as the context for the business rules.

4. One of the Classes represents the actual application of the rules; under this Class is a Rule Flow Activity and Activity diagram, in which you create the Rule Tasks under which the business rules are grouped, in the sequence in which the business rules are executed. If you

require a higher level of grouping to define different areas of rule application, you can create other rule-processing Classes with their own Activities.

5. Return to the Business Rules diagram and drag in the Rule Task elements from the Browser window, assigning to each Task the corresponding Business Rule elements.

6. Compose and validate the business rules, using the Rule Composer.

7. If you want code that applies and executes the business rules, generate it from the Class elements that contain the Business Rules Activities.

The steps are represented graphically in this flow:
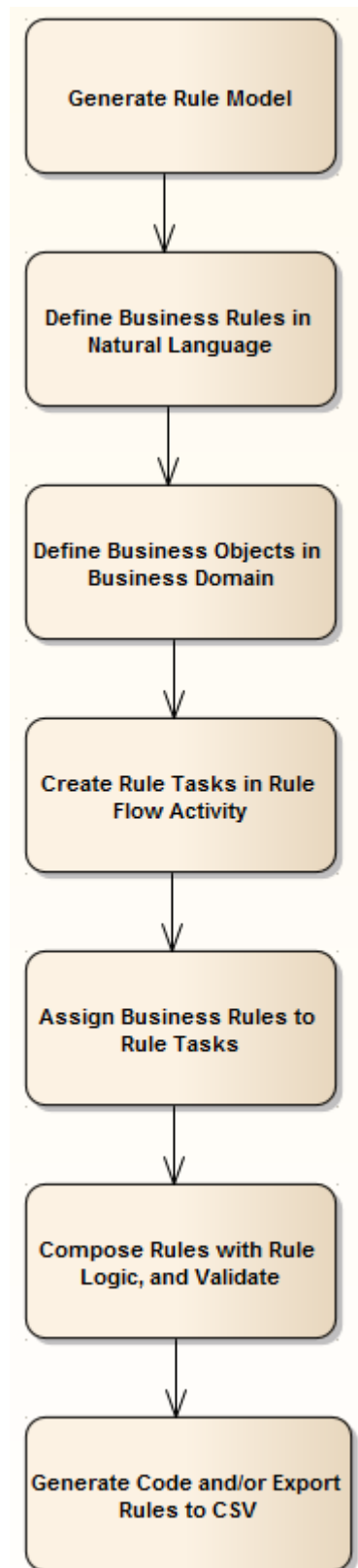
Notes

- Business Rule Modeling is available in the Unified edition and the Ultimate edition of Enterprise Architect

Generate a Business Rules Model

The Business Rules Model captures:

- The rules that apply to a business process
- The business tasks or objects that the rules take effect on, and
- The actual processing that takes place to apply the rules and produce a decision or result

As a very useful starting point in setting up your Business Rules model, you can generate the model structure and initial components using the Enterprise Architect Model Wizard.

Access

Use any of the methods outlined here to open the Model Wizard window.

Ribbon	Design > Model > Add > Model Wizard
Context Menu	Right-click on Package or Model Root node | Add a Model using Wizard
Keyboard Shortcuts	Ctrl+Shift+M

Other	Browser window header bar |

Generate Business Rules Model from Model Wizard

Step	Action
1	Select the 'Model Patterns' tab.
2	Click on the Perspective drop-down arrow and select 'Requirements | Business Rule Model'.
3	Click on the 'Starter Business Rule Model' option. If you prefer, you can read the information on this model in the right-hand panel.
4	Click on the Create Pattern(s) button. A Business Rules Model structure is generated in the Browser window, as shown.
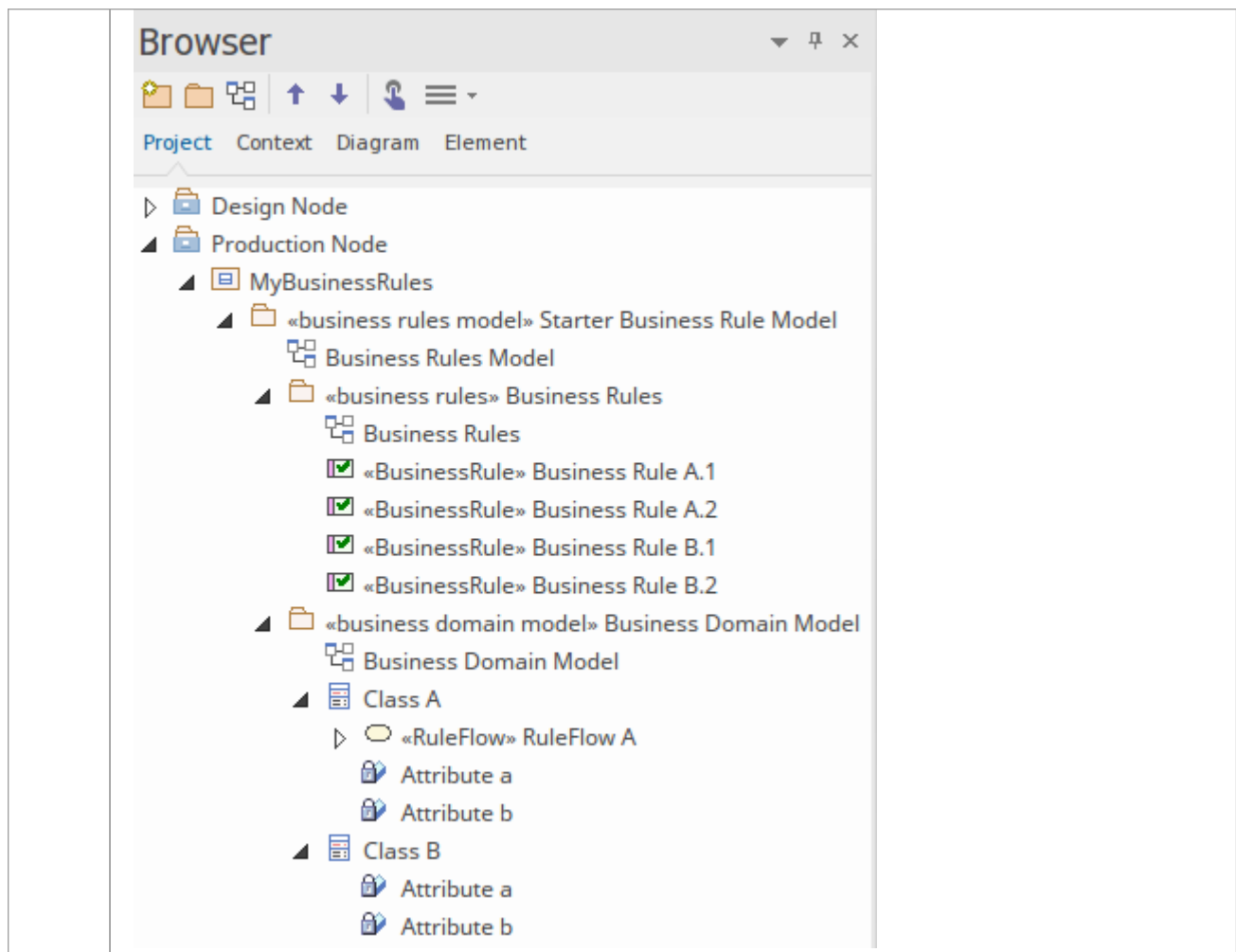
Example Diagram

This Business Rules Model diagram was generated at the top level of the model and encapsulates the components of the Business Rules model.

Business Rule Model

Business Domain Model
- + Application
- + Car
- + Customer
- + Rent
- + RentalSystem
- + CarType

Defining Business Rules
- ☑ + Car must not be rented to Customers of age less than 18
- ☑ + Car must not be rented to Customers with Bad History level 3
- ☑ + Car must not be rented to customers without a valid license number
- ☑ + Determine Penalty
- ☑ + Penalty must not be applied for Customers with Bad History level 0
- ☑ + Penalty of 10 % of rent must be applied for Customers with Bad History Level 1
- ☑ + Penalty of 20 % of rent must be applied for Customers with Bad History Level 2
- ☑ + Rent for AWD cars is 100 AUD per day
- ☑ + Rent for Luxury cars is 150 AUD per day
- ☑ + Rent for Small cars is 80 AUD per day
- ☑ + Rent Payable is calculated as the product of RentPerDay and RentalPeriod in days
- ☑ + Total Amount Payable is calculated as the sum of Rent Payable and Penalty if any.

Notes

- Business Rule Modeling is available in the Unified edition and the Ultimate edition of Enterprise Architect

- To see an example of a Business Rules model, open the EAExample model provided with the installer, and expand:

    Example Model > Analysis and Business Modeling > Business Domain Model > Business Rule Model > Business Domain Model   and > Defining Business Rules

Model Business Rules

In the Business Rules Model, you initially define each business rule as a Business Rule element and later group these rules by linking them (using Dependency connectors) with Rule Task elements. In the:

- First stage you assemble a collection of rules
- Second stage you organize the rules into groups and sequences through the Rule Task elements created in the Business Domain model, and refine them by adding further Business Rule elements (and, if appropriate, further Rule Task elements)

In the Car Rental example in the EAExample model, rules have been defined and grouped to perform an eligibility check on a customer, to determine if the customer is eligible to rent a car.

Define Business Rule elements

Step	Action
1	In the generated Business Rules Model, expand the Business Rules Package and double-click on the Business Rules diagram to open it. The diagram shows two example Business Rule elements connected to an example Rule Task.
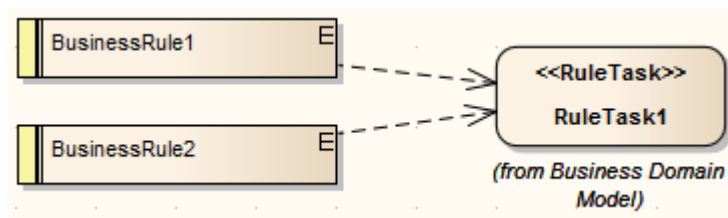
In the Diagram Toolbox, the 'Rule Model' pages display.
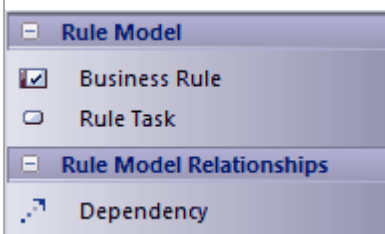
2	If you should need to create another (empty) Business Rules diagram, right-click on the Business Rules Package and select 'Add Diagram'.
	On the 'New Diagram' dialog, type an appropriate name in the 'Diagram' field, click on the 'Select From' header and select the 'Project Management > Complete' Perspective group, and then scroll down and select the Business Rule Model Perspective from the panel beneath the header.
	Select 'Rule Model' in the 'Diagram Types' panel and click on the OK button.
3	For each business rule you want to identify, drag the 'Business Rule' icon from the Toolbox.
	Type the rule (or a shortened version of it) in the

		'Short Description' field of the element 'Properties' dialog. This displays as the name of the element in the diagram. You will later define the parameters of the rules using the Rule Composer.
4		When you have created all the required Business Rule elements, create the Business Domain model, the Rule Flow Activity diagram and the Business Task elements.

Associate Business Rules with Business Tasks

After you have created the Business Domain Model and Rule Flow Activity (which contains the Rule Task elements in the context of the business process), you can link the Business Rule elements to the Rule Task for the action in which the rules take effect. You can apply a Business Rule to more than one Rule Task, if it has an effect in different contexts.

There are several ways to establish the relationship between Business Rule and Rule Task.

Method	Action
Use the Business	Open the Business Rules diagram and drag onto it a Rule Task element from the

Rules diagram	RuleFlow Activity in the Browser window.
	Create a Dependency relationship to the Rule Task from each Business Rule element that applies to it (by dragging the connector from the Toolbox, or by using the Quick Linker arrow on the Business Rule element).
	Repeat the process for the next Rule Task element from the Rule Flow Activity.
	Create any additional Business Rule and Rule Task elements that are necessary (this should not be a common event); you must add any new Rule Tasks to the Rule Flow Activity diagram.
	Save the diagram and open the Rule Composer for the first Rule Task element.
Assemble existing elements through the Rule Composer	In the Browser window, open the Rule Composer on a Rule Task element, and drag each applicable Business Rule element from the Browser window into an empty row of the Rule Statements table.
	This establishes a Dependency relationship between the Rule Task element and each Business Rule element.

	Continue to compose the rule conditions.
Use the Relationship Matrix	You can quickly create all the Dependency relationships between a number of Rule Task elements in the Business Domain Model Package and the Business Rule elements that apply to each one.
	Set the source element 'Type' to 'BusinessRule' and the target element 'Type' to 'RuleTask'.
	Set 'Link Type' to 'Dependency', and 'Direction' to 'Source->Target'.
	Set the 'Source Package' to 'Business Rules' and the 'Target' to 'Business Domain Model'.
	In the cell at each intersection of a Business Task column and the appropriate Business Rule row, right-click and select the 'Create new relationship	UML:: Dependency' option.
	When you have finished creating the relationships, close the Relationship Matrix.
	Open the Rule Composer for the first Rule Task element in the Browser window.

Example

However you connect the Rule Task and Business Rule elements, if they are added to a diagram it might display as shown.
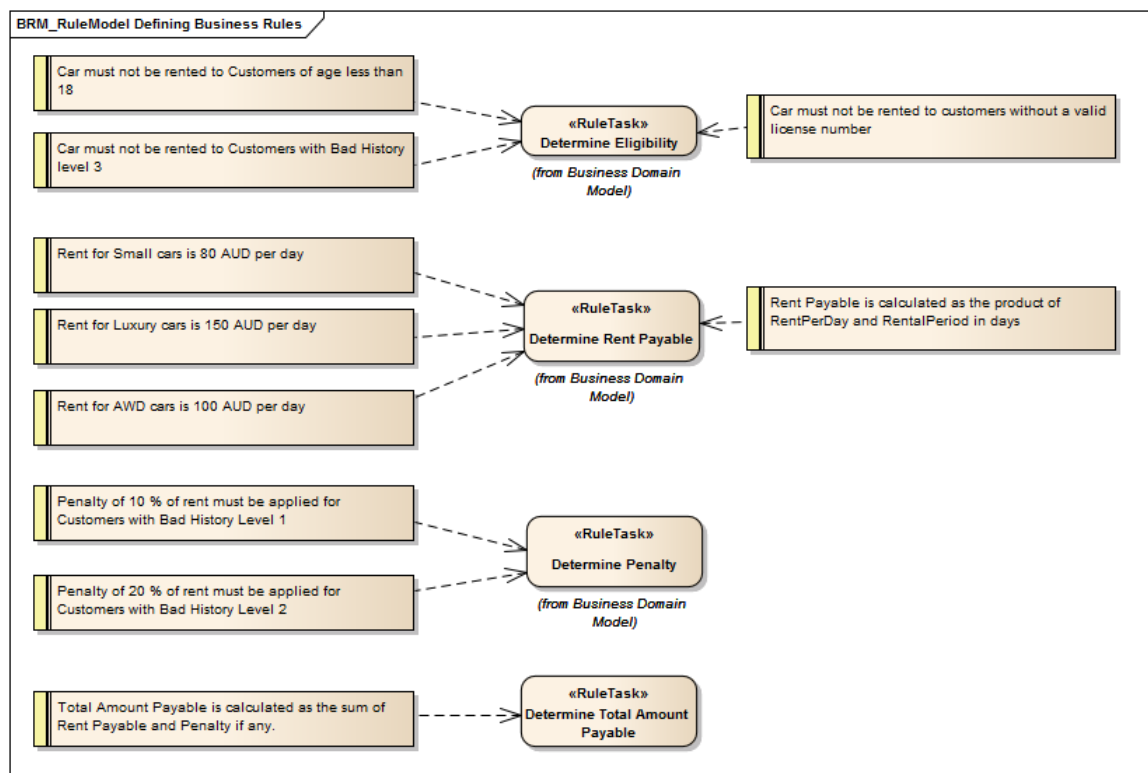
Notes

- Business Rule Modeling is available in the Unified edition and the Ultimate edition of Enterprise Architect

Create a Business Domain Model

The Business Domain Model provides the business vocabulary - the terms and facts - on which Business Rules can be modeled. In Enterprise Architect a Business Domain model is represented as a conceptual Class diagram, as illustrated by this diagram from the Car Rental System model from the EAExample model.
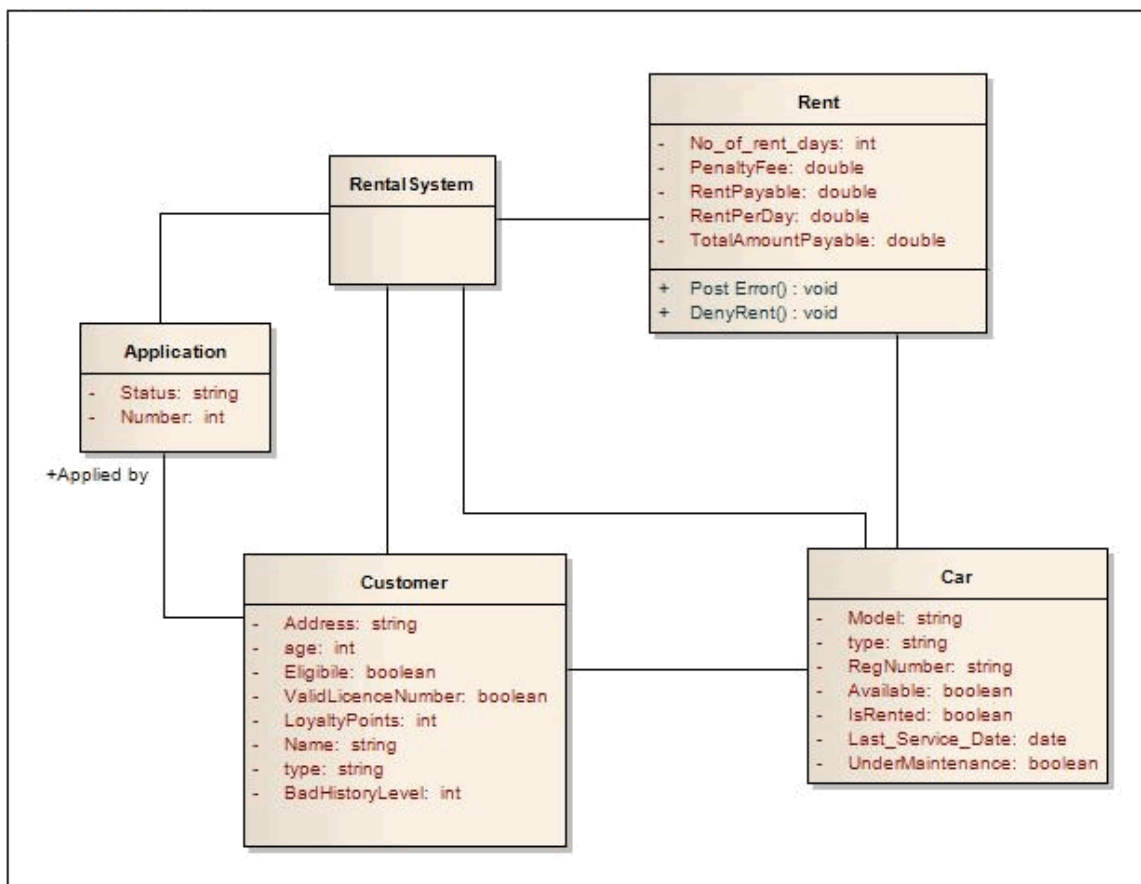
In the Business Domain model shown in this diagram, the Classes Rent, Customer, Car and Application, together with their attributes and operations, provide the terms for the business vocabulary for the car rental system. The operations and attributes identify the conditions that must be met, the actions that must be taken, and the calculations that

must be made to filter and apply the rules to provide a specific value or outcome.

The Class Rental System processes the rules; to make this possible, you add a Rule Flow Activity as a behavior for this Class.

When you create a Rule Flow Activity under a Class, you model the events and sequence as a structure of Rule Tasks (Actions). When you generate code for the Class (in the example, Rental System) the rule flow behavior is rendered as a method inside the Class.

Alternatively, if you have existing operations in the Class that already suit the purpose, you can model business tasks in those operations. When code is generated for the Class, the rules logic is generated as the method body for the corresponding operation.

Notes

- Business Rule Modeling is available in the Unified edition and the Ultimate edition of Enterprise Architect
- When you create Classes in the Business Domain model, select the correct language for code generation to ensure that the correct data type is set for attributes and operation parameters
- Business Rules code generation is supported for these languages:
    - C++
    - C#

- Java
- VB.Net

Create a Rule Flow Activity

When you set up a Business Domain model within a Business Rules model, you create a Rule Flow Activity as a behavior for one of the domain Classes, to enable that Class to process a set of rules. In the Rule Flow Activity you create a number of Rule Task elements, which are stereotyped Actions that group Business Rules for a specific task. The Rule Flow Activity automatically generates a Rule Flow diagram, in which you create the Rule Task elements and model the sequence in which they are executed.

Add a Rule Flow Activity to a Class
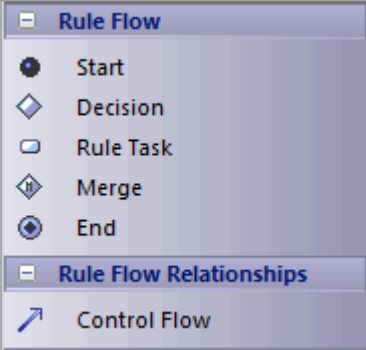
Step	Action
1	On the Business Domain model diagram, right-click on the Class that processes the rules (in the Car Rental example in the EAExample model, this would be Rental System).
2	From the context menu select the 'New Diagram | RuleFlow Activity' option. A new Rule Flow Activity is created with a Rule Flow diagram, which immediately opens. Go on to create the Rule Task elements.

	Code generation for a Rule Flow model renders each RuleFlow Activity as a set of operations or methods. Depending on what these methods are to do, you might want to pass in parameters to be used within the Rule Flow Activity.

Add Rule Task elements to the Rule Flow Activity diagram

You can create Rule Task elements directly under the Rule Flow Activity in the Browser window, by clicking on the 'New Element' icon in the Toolbar and selecting the UML::Activity toolset, Action element type and RuleTask stereotype. However, it is much simpler to create the elements on the Rule Flow diagram, and at the same time to organize them into their processing sequence.

Ste p	Detail
1	(If necessary) Click on 🔍 to display the 'Find Toolbox Item' dialog and specify 'Rule Flow'.

2	Drag the 'Rule Task' icon from the Toolbox onto the diagram and give the element, as a name, the title of the task it represents, such as Calculate Debit Charge or Determine Eligibility. Create a Rule Task element for each task or action in the process. You can also use the Quick Linker arrow to create the new elements and Control Flow connectors.
3	Organize the Rule Tasks into a sequence of events, initiated and terminated by the Start and End elements, and representing any branching and rejoining with Decision and Merge elements. All elements are connected by Control Flow connectors. See the example diagram.
4	Go to the Business Rules model diagram and group the Business Rule elements on their appropriate Rule Task element.

Example

This Rule Flow diagram is from the EAExample model Car Rental example.
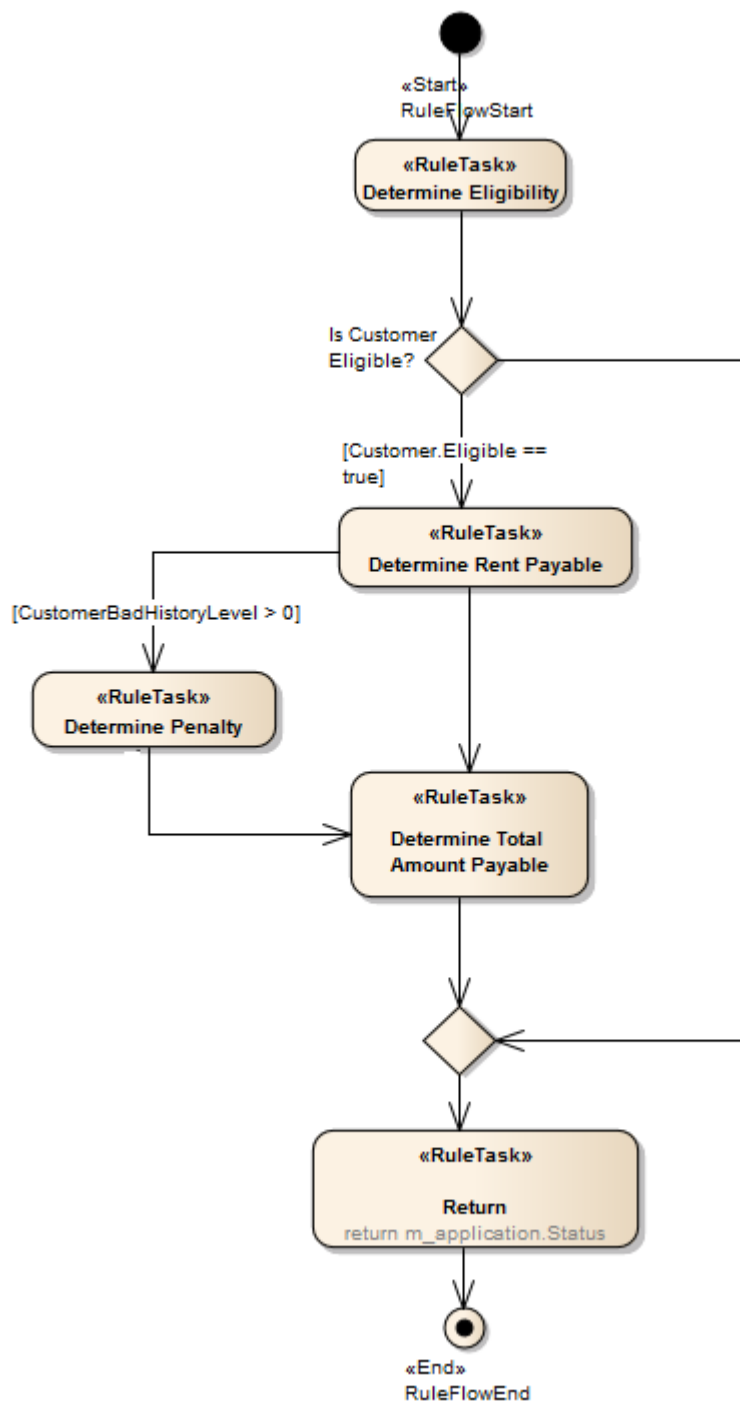
The Rule Task elements Determine Eligibility, Determine Rent Payable, Determine Penalty and Determine Total Amount Payable group the business rules for the specific task indicated by the element name.

Notes

- Business Rule Modeling is available in the Unified edition and the Ultimate edition of Enterprise Architect
- In a Rule Flow diagram, every Decision node has a matching Merge node to ensure proper code generation
- For code generation, the Rule Task elements must be grouped inside the appropriate Rule Flow Activity in the Browser window (Business Rule elements can be defined anywhere in the model, as they can be used in more than one Rule Task)

Pass Parameters to Rule Flow Activity

When you generate code for a Rule Flow model, each RuleFlow Activity is rendered as a set of operations or methods. You can pass in Activity Parameters to be used by the Rule Tasks within the Rule Flow Activity, to define what you want the methods to do. You can use the parameters as condition variables or action variables in the Business Rule Decision Table, or as rule variables in the Computation Table for any of the Rule Tasks.

If the Activity Parameter is not accessible to a Rule Task, the system displays an error message.

Access

Context Menu	Browser window | Double-click Rule Flow Activity element > Parameters

Define parameters to be used within a Rule Flow Activity

Step	Action

1	On the 'Parameters' page, create and define each parameter, in particular the 'Type' and 'Default' values.
2	Save each parameter and, when you have finished setting all parameters, close the 'Properties' dialog.

Example

In this hierarchy, the parameters m_rent and m_application can be used by any of the Rule Tasks under the ProcessApplication Rule Flow Activity.
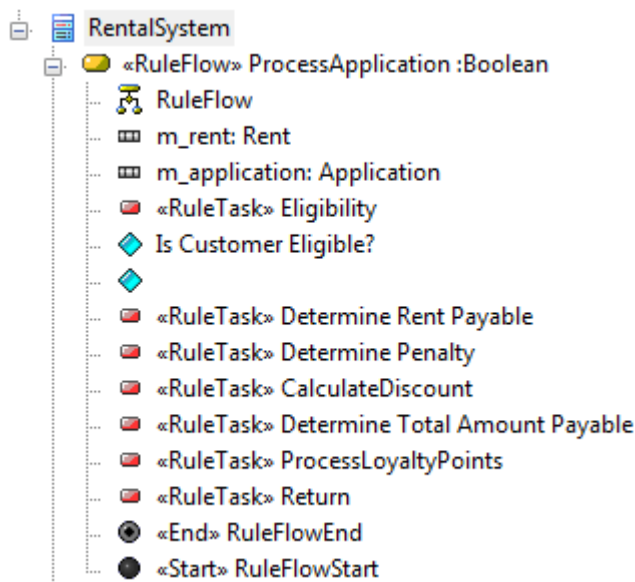
Model Rules In an Operation

You can model business rules as Business Rule elements in the Business Rules Package, and attach them to the Rule Task elements in a RuleFlow Activity diagram.

Alternatively, in the Business Domain model, if you have operations in the rules processing Class that represent business action, you can define each of those operations as a Rule Task and attach the Business Rules to these operations on the Business Rules diagram or within the Rule Composer.

Access
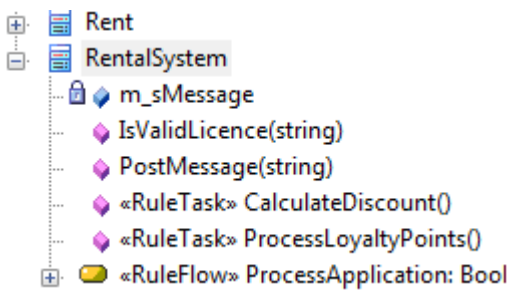
Context Menu	Browser window| Double-click on Operation > General

Model Business Task on an operation

Step	Action
1	In the 'Stereotype' field type 'RuleTask'.

2	Click on the Save button and on the Close button.
3	Drag the Operation from the Browser window onto the RuleFlow Activity diagram. Also assign Business Rules to the operation, as for a Rule Task element.

Representation of Operation Rule Tasks

View	Detail
Browser Window	The operations stereotyped as RuleTask display in the Browser window as shown: To pass the parameters for these operation calls, open the operation 'Properties' dialog and select the 'Call' page, then set the 'Behavior field to the operation to be called; under the 'Arguments' field, click on the Edit button and set up or edit the argument values to

	be passed.
	On code generation, the code for rules logic is generated in the method body.
On the Rule Flow Diagram	When you drag and drop a RuleTask-stereotyped operation onto a Rule Flow diagram, an operation Call Behavior Action is created, as shown by the two darker elements.
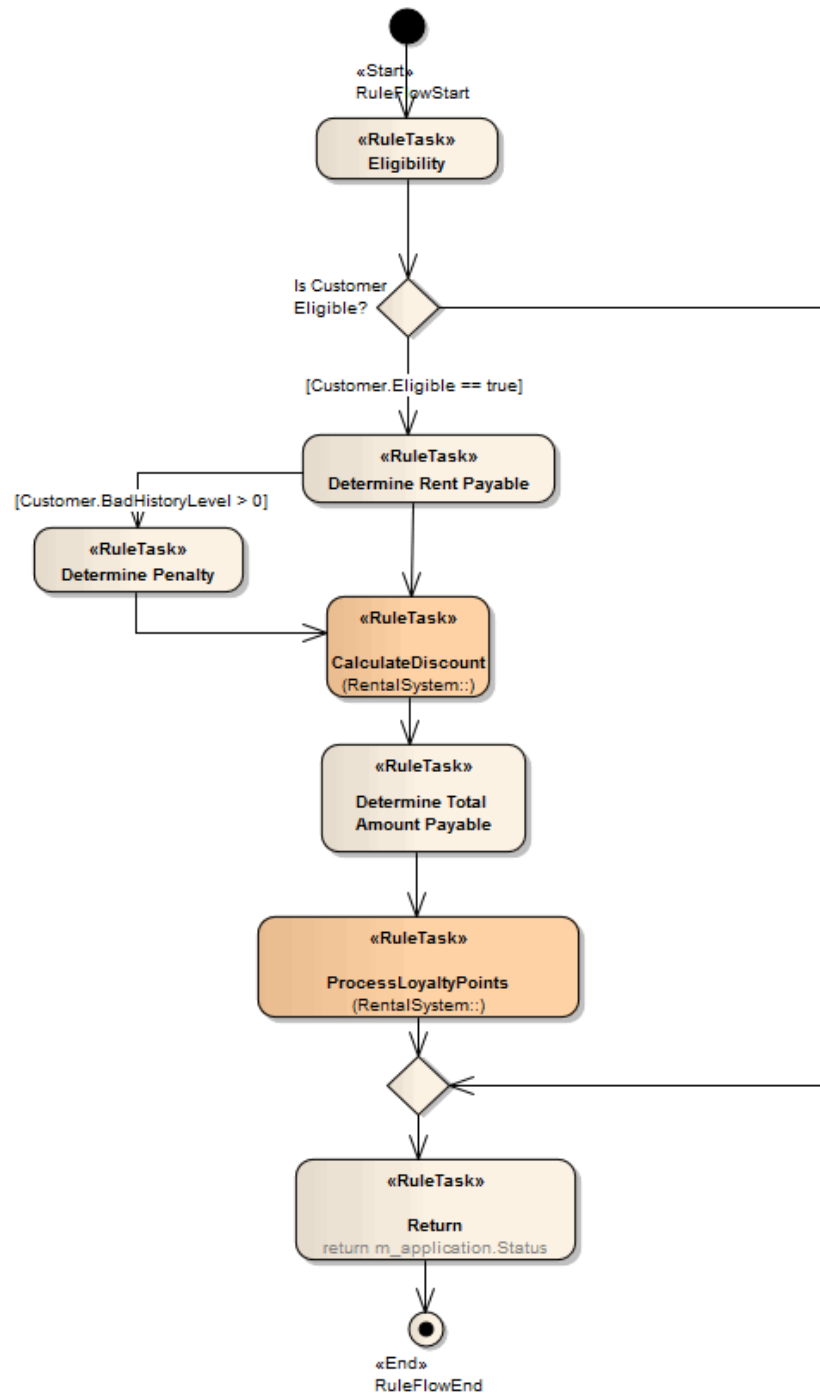
You must build the Call Behavior Actions into the activity flow, as for the normal Rule Task Actions.

Notes

- Business Rule Modeling is available in the Unified edition and the Ultimate edition of Enterprise Architect

Compose Business Rules

In modeling your Business Rules, you first define each business rule as a conceptual-level, plain text string within a Business Rule element, and then group the rules by association with Business Task elements. Your next step is to define exactly how the rules operate within the task, setting up the values, conditions, actions and computations that define the actions of a single rule or a combination of rules. For this you use the Rule Composer, with which you transform each conceptual-level business rule statement into a logical level, technology-specific tabulated statement that you can either:

- Generate code from or
- Download to a spreadsheet application such as Microsoft Excel, via a CSV file

Access

Context Menu	Browser window or open diagram | Right-click on a Rule Task element | Rule Composer

Rule Composer Tables

The Rule Composer displays as a view in the central work area, divided into three tables.

Table	Detail
Rule Statements	The Rule Statements table lists the rules associated with the selected Rule Task; you add a rule to the table by dragging an existing Business Rule element from the Browser window onto an empty row of the table. You do not create new rules within the table.
Decision	The Decision Table is used to model conditional rules (for example: Cars must not be rented to customers of age less than 18). The table has three sections: • Rule Conditions – to model condition variables • Rule Actions – to model action variables • Rule Bindings – to link the rule in the rule table
Computation Rule	Using the Computation Rule table, you model rules that require a calculation to

	be performed on the source information, and/or the interaction of rules. The table has these columns: • Computation Rule Actions • Expression • Rule Bindings • Rule Dependency

Notes

- Business Rule Modeling is available in the Unified edition and the Ultimate edition of Enterprise Architect

- To assist with traceability as you complete the relationships across the Rule Composer, selecting an entry in one table automatically highlights the corresponding rows and columns of the other tables; for example, if a Rule Statement is selected, the related rule column in the Decision Table and row in the Computation Rule Table are highlighted
Similarly, if a Computational Rule is selected, the corresponding column in the Decision Table and row in the Rule Statements list are highlighted

- If the table columns are not wide enough to display their contents, you can drag the column header margins to increase the width; all tables on the 'Decision Table' tab and the 'Computation Rule' Tab are linked, so if you

increase the column width on one table it changes the width on all tables

- The Rule Composer can be opened and the rules logic viewed in the Lite edition of Enterprise Architect

Add and Remove Rules

When you open the Rule Composer for a Business Task, all Business Rules currently linked to that task are already listed in the Rule Statements table. You can add further rules that are available in the Browser window, or remove a rule from the table that is no longer applicable to the task.

You do not create or delete the rules themselves in the Rule Composer.

Access

Context Menu	Browser window or open diagram | Right-click on a Rule Task element | Rule Composer

Add a business rule to be associated with the selected Rule Task

When you add a Business Rule to the Rule Statements table, if the Business Rule element is not already on the Business Rules diagram, the element is added to the diagram. A Dependency relationship is created between the 'new'

Business Rule and the selected Rule Task.

When you open the Rules Composer	Action
There are no rules listed	The Rule Statements table has a single empty row. Select the Business Rule in the Browser window and drag it onto this empty row. Another empty row is automatically created underneath the first row. Drag the next Business Rule from the Browser window onto the Rule Statements table.
There are rules listed	The 'Rule Statements' table has no empty rows. Click on the 'No' column and select the 'Add Row' context menu option; an empty row is added to the table. Drag a Business Rule from the Browser window onto the table.

Remove a rule that is no longer required in the Rule Composer

Right-click on the appropriate 'No' field and select the 'Remove Rule' option. The rule is removed from the Rule Composer and the Dependency relationship with the Rule Task element is deleted.

The Business Rule element is not deleted from either the Business Rule diagram or the Browser window (where, in either case, it might be in use with other Rule Task elements).

Define Rule Conditions

When you create the Business Domain model, you set up a number of Classes that define the business terms and entities (such as Customer) and their associated attributes and operations. You create the attributes and operations as you set up the Class, with at least some values or parameters, and you tailor some of the features to the rules to define the conditions through which a rule takes effect, in the Rule Conditions table.

For example, in a Class that defines the properties of Car, the attribute Type might be used to set the condition 'Car is: Small, Medium or Large', through which the rules defining base rental charge would be filtered and applied.

Access

Context Menu	Open diagram	Right-click on a Rule Task element	Rule Composer > Decision Table

Model Rule Conditions

Ste p	Action
1	From the appropriate Class element in the Browser window, drag and drop the condition attribute (such as age) or operation (such as IsValidLicense()) onto the first empty field in the 'Rule Conditions' column. • The 'Rule Conditions' fields apply Intelli-sense to display possible entries for the field; press Ctrl+Space in the field to display a list of possible Classes, double-click on the selected Class, type . (a period) and double-click on the attribute or operation in the automatically-displayed list • If the 'Rule Condition' is of type enum, the 'Allowable Values' field is automatically set with the enum literals; the procedure then ends here
2	Determine a range of accepted values for the Rule Condition.
3	Right-click on the 'Allowable Values' column and select: • For an attribute, the 'Edit Allowable Values' option; the 'Edit Allowable Values' dialog displays • For an operation, the 'Edit Parameters' option; the 'Edit Parameters' dialog displays (see step 5) Type each required value or range of values in the 'Value' field, and click on the Save button to display

	the value in the 'Allowable Values' list box *age* could have the values: • <18 • >18 and <50 • >50 *IsValidLicense()* could return: • True • False
4	Click on the OK button to save the values and close the dialog; for an attribute, a new constraint AllowableValues is created. • You can check this constraint by opening the 'Properties' dialog for the attribute and selecting the 'Constraints' tab • If the Rule Condition references an enumeration, the enum literals are not editable in the 'Edit Allowable Values' dialog
5	If the Rule Condition is an operation, you can pass parameters to it. Right-click on the 'Allowable Values' field, and select the 'Edit Parameters' option; the 'Edit Parameters' dialog displays, listing the parameters that already exist for the operation. Select the parameters and type their values into the 'Value' text box; click on the OK button to cancel the

	dialog.
	• You can add an operation as a Rule Condition more than once, so you can call the operation with different sets of parameters
6	To add another Rule Condition, right-click on the 'No' column and select the 'Add Row' option; an empty row is added to the table.
	To remove a Rule Condition from the table, right-click on the appropriate 'No' field and select the 'Delete Row' option.
	• If the condition is based on an attribute, this does not affect either the original attribute or the new constraint in the model; you can either re-use the attribute with its constraint, or use the attribute 'Properties' dialog to remove the constraint
7	If any of the condition values invoke an action or a decision, you can define it as a Rule Action.

Define Rule Actions

In the Classes in the Business Domain model, the attributes and operations that you set up to define the conditions through which a rule takes effect might in their turn call a decision attribute or an operation, which you can assign as an action in the Rule Actions (Outcome) table. You create the attributes and operations as you set up the Class, with at least some values or parameters, and you tailor some of these features to define the actions that determine the effect of the rule.

Access

Context Menu	Open diagram | Right-click Rule Task element | Rule Composer > Decision Table

Model Rule Actions

Step	Action

1	From the appropriate business term Class element in the Browser window, drag and drop the decision attribute (such as Eligible - Yes/No) or operation (such as post error) onto the first empty field in the 'Rule Actions (Outcome)' column. • The 'Rule Actions' fields apply Intelli-sense to display possible entries for the field; press Ctrl+Space in the field to display a list of possible Classes, double-click on the selected Class, type . (a period) and double-click on the attribute or operation in the automatically-displayed list
2	For an attribute, if the dropped action variable is of type 'enum', the 'Allowable Values / Parameters' fields are automatically set with the enum literals. Otherwise, double-click on the 'Allowable Values / Parameters' field (or right-click and select the 'Edit Allowable Values' option); the 'Edit Allowable Values' dialog displays. Click on the New button, type a possible value in the text box (such as 'Yes' or 'Accept'), click on the Save button, and repeat the process for the next possible value (such as 'No' or 'Reject').
3	When you have entered all possible values, close the dialog and select the appropriate response in the 'Result<n>' column field underneath the appropriate 'Value<n>' field for the calling condition.

4	For an operation, a checkbox displays in each of the 'Result<n>' column fields; to call the operation as the response to a value of the corresponding condition, select the checkbox in the 'Result<n>' field underneath that condition 'Value <n>' field.
5	To pass parameters to the operation, double-click on the 'Allowable Values/Parameters' field (or right-click and select the 'Edit Parameters' menu option); the 'Edit Parameters' dialog displays.
	Select the parameters and type the values into the 'Value' text box; click on the Save button and close the dialog.
	You can add an operation as a Rule Action more than once, so you can call the operation with different sets of parameters.
6	Click on the Save button in the Rule Composer toolbar to save the values.
7	To add another Rule Action, right-click on the 'No' column and select the 'Add Row' option; an empty row is added to the table.
	To remove a Rule Action from the table, right-click on the appropriate 'No' field and select the 'Delete Row' option; this does not affect the original attribute or operation in the model.

Bind Rules to Conditions and Actions

After defining which rules are applied, which conditions are evaluated and/or which actions are taken in the completion of a Rule Task, you bind the rules to the conditions and actions to define:

- Which rules are evaluated against a condition
- What values of that condition are applicable to each rule
- What decisions or actions are taken on each value output from evaluation of the rule, and what the result of that decision or action is

You bind the rules, conditions and actions using the 'Rule Bindings' table, just above the 'Rule Conditions' table.

Access

Context Menu	Open diagram | Right-click on a Rule Task element | Rule Composer > Decision Table

Bind rules

Ste	Action

p	
1	Click on the 'Rule Bindings' field immediately above the appropriate 'Value<n>' and/or 'Result<n>' field. Click on the number of the rule that is evaluated against the condition for the specific value and to generate the action to provide the specific result identified in those two fields.
2	Ensure that the values set in the 'Value<n>' and/or 'Result<n>' field underneath the rule number do all operate under the rule.
3	Click on the Save icon in the Rule Composer toolbar. Continue to bind business rules to conditions and actions having logical values for an instance of the rule.

Example

- Rule 2 is 'A car must not be rented to a customer of age less than 18'
- The first Rule Condition is Customer.age
- The 'Value1' field contains the value '< 18' against that Rule Condition

- The first Rule Action (or decision) is Customer.Eligible
- The Result1 field contains the value 'No' against that Rule Action
- The second Rule Action is Application.Status
- The 'Result1' field contains the value 'Reject' against that Rule Action

You would select '2' in the 'Rule Bindings' field over the 'Value1/Result1' column, so that a when a customer's application to rent a car is tested against rule 2, if the customer is found to be under 18 they are automatically classified as ineligible and the application rejected.

Define Computation Rules

In a business process, some actions and effects are dependent on the interaction of rules with either another rule or a calculation. For example:

- Rule 1) Cars cannot be rented to customers aged under 18 years, Rule 2) Customers must have a valid driving licence - Rule 1 must be satisfied before Rule 2 is tested

- Rule 4) Penalty of 10% of total rental cost applies to customers with Bad History Level 1 - If the customer has a Bad history Level 1, multiply the rent payable value by 1.1

Access

Context Menu	Open diagram | Right-click on a Rule Task element | Rule Composer > Computation Rule Table

Define a computation rule

Step	Action

1	In the Browser window, expand the appropriate business entity Class in the Business Domain model, and drag the attribute that represents the computed action into the 'Computation Rule Actions' field. Alternatively, you can use Intelli-sense in both the 'Computation Rule Actions' field and the 'Expression' field to display a list of possible Classes and attributes: 1. Press Ctrl+Space in the field to display the list of Classes. 2. Double-click on the required Class, and type . (period) after it; a list of attributes for that Class displays automatically. 3. Double-click on the required attribute.
2	In the 'Expression' field, complete the expression to be evaluated. For example: If computing Rule 4, and you have selected *Rent.RentPayable* in the 'Expression' field, then type **1.1* immediately after it.
3	In the 'Rule Bindings' field, click on the drop-down arrow and select the number of the rule being modeled (as listed in the Rule table) to link the computation data to the rule.
4	If the rule depends on another rule being satisfied first (as for Rule 2), type the number of that rule in

	the 'Rule Dependency' field.
5	Click on the Save icon in the Rule Composer toolbar to save the computation rule. If the computation rule is also a Rule Conditions rule, add the condition variable in the Decision table and bind the appropriate rule in the 'Rule Bind' section.

Validate Business Rules

Before you generate code for the Rule Task elements, or export them to a CSV file, it is recommended that you validate the business rules in the Rule Composer. When you do this, the business rules on the Rule Composer are parsed and any errors or warnings that might indicate incomplete or unfavorable code generation are displayed on the 'Rule Composer Validation' tab on the System Output window.

Access

Context Menu	Open diagram	Right-click on a Rule Task element	Rule Composer

Validate Rules

Step	Detail
1	Click on the 'Validation' icon (green tick) in the Rule Composer toolbar:

		The System Output window displays, showing the status of the validation - Validation complete, plus any validation errors or warnings - on the 'RuleComposer Validation' tab.
	2	Double-click on each warning or error message in turn to highlight and investigate the faulty data that caused that message, in the Rule Composer.

Export Composed Rules to CSV

It is possible to export composed Business Rules to an external spreadsheet - such as Microsoft Excel - as a .csv file. You can then maintain the spreadsheet in that tool for use as a reference for the organization, if you have not created an application to automatically apply those rules.

Access

Context Menu	Open diagram | Right-click on a Rule Task element | Rule Composer

Export the contents of the Rule Composer to a CSV file

Step	Action
1	Click on the 'Export to CSV' icon (▦) in the Rule Composer toolbar. The 'Windows Browser' dialog displays.

2	Browse to the required file location and type in a name for the .csv file to export the data into.
3	Click on the Save button to export the data.

Code Generation For Business Rules

After you have modeled rules for all Rule Task elements in the Rule Flow diagram, the Business Domain model is ready for code transformation. The internal code templates provided for generating technology-specific rule code work hand-in-hand with the EASL code templates to generate the code from the rules-processing Class and its Rule Flow structure.

Return a value from the Rule Flow behavior

Step	Action
1	Double-click on the last Rule Task element before the end node of the Rule Flow diagram. The element's 'Properties' dialog displays.
2	Click on the 'Effect' tab.
3	In the 'Effect' field, type the return statement; for example, return True.
4	Click on the Save button, and on the OK button to close the dialog. Generate code for the Class containing

	the rule flow behavior (in our initial example, Rental System); the code for business rules logic is generated, with the rule statements expressed in natural language as comments.

Example

This code snippet was generated from the Rental System Class element in the EAExample model:

```
/////////////////////////////////////////////////////////
// RentalSystem.cs
// Implementation of the Class RentalSystem
// Generated by Enterprise Architect
// Created on: 26-July-2016 2:39:23 PM
/////////////////////////////////////////////////////////
    using System;
    using System.Collections.Generic;
    using System.Text;
    using System.IO;

    public class RentalSystem
    {
        public Customer m_Customer;
```

```
public Car m_Car;
public Rent m_Rent;

public RentalSystem()
{
}
~RentalSystem()
{
}
public virtual void Dispose()
{
}
/* Begin - EA generated code for Activities and
Interactions */
public bool ProcessApplication(Rent
m_rent,Application m_application)
{
    // behavior is an Activity
    /*CAR MUST NOT BE RENTED TO
CUSTOMERS WITHOUT A VALID LICENSE
NUMBER*/
    if(m_Customer.ValidLicenseNumber ==
"FALSE")
    {
        m_application.Status = "Reject";
```

```
            m_Customer.Eligible = false;
      }
      /*CAR MUST NOT BE RENTED TO
CUSTOMERS OF AGE LESS THAN 18*/
      if(m_Customer.age < 18)
      {
          m_application.Status = "Reject";
          m_Customer.Eligible = false;
      }
      /*CAR MUST NOT BE RENTED TO
CUSTOMERS WITH BAD HISTORY LEVEL 3*/
      if(m_Customer.BadHistoryLevel == 3)
      {
          m_application.Status = "Reject";
          m_Customer.Eligible = false;
      }
      if (Customer.Eligible == true)
      {
          /*RENT FOR SMALL CARS IS 80 AUD PER
DAY*/
          if(m_Car.type == Small)
          {
              m_rent.RentPerDay = 80;
          }
          /*RENT FOR AWD CARS IS 100 AUD PER
```

DAY*/

```
        if(m_Car.type == AWD)
        {
            m_rent.RentPerDay = 100;
        }
```

/*RENT FOR LUXURY CARS IS 150 AUD PER DAY*/

```
        if(m_Car.type == Luxury)
        {
            m_rent.RentPerDay = 150;
        }
```

/*RENT PAYABLE IS CALCULATED AS THE PRODUCT OF RENTPERDAY AND RENTALPERIOD IN DAYS*/

```
        m_rent.RentPayable = m_rent.RentPerDay * m_rent.No_of_rent_days;
        if (CustomerBadHistoryLevel > 0)
        {
```

/*PENALTY OF 20 % OF RENT MUST BE APPLIED FOR CUSTOMERS WITH BAD HISTORY LEVEL 2*/

```
            if(m_Customer.BadHistoryLevel == 2)
            {
                m_rent.PenaltyFee = m_rent.RentPayable * 0.2;
            }
```

```
                    /*PENALTY OF 10 % OF RENT MUST
BE APPLIED FOR CUSTOMERS WITH BAD HISTORY
LEVEL 1*/

                    if(m_Customer.BadHistoryLevel == 1)
                    {
                        m_rent.PenaltyFee = m_rent.RentPayable
* 0.1;
                    }
                }
                else
                {
                }
                    /*TOTAL AMOUNT PAYABLE IS
CALCULATED AS THE SUM OF RENT PAYABLE
AND PENALTY IF ANY.*/

                    m_rent.TotalAmountPayable =
m_rent.RentPerDay + m_rent.PenaltyFee;
            }
            else
            {
            }
            return m_application.Status;
        }
        /* End - EA generated code for Activities and
Interactions */
    }
```

//end RentalSystem

Notes

- Business Rule Modeling is available in the Unified edition and the Ultimate edition of Enterprise Architect

Business Models

Business analysis, through the efforts of a number of international organizations, has become a rigorous and regulated discipline. The Business Analyst is expected to discover, analyze and synthesize large volumes of information and be able to present it clearly and meaningfully to heterogeneous audiences. The Business Analyst typically creates models of requirements, policies, business rules, business processes and information. All of these concepts can be created easily in a single repository and integrated, allowing the analyst to spend their time on analysis instead of copying content between tools. At any stage during the analysis, high quality documentation tailored to a specific audience and spanning all the content can be created in a wide range of formats.

Business Model Diagram Types

Diagram Type	Description
Requirements	Create, model and manage requirements directly in line with your other project tasks. Trace from high level requirements to deployment artifacts.
Use Cases	Describe the functional requirements of

	the system, the manner in which outside things (Actors) interact at the system boundary, and the response of the system.
BPMN	Capture the behavior and the information flows within an organization or system.
Decision Models	Define decisions with models and tables using the decision trees or decision tables.
Mind Mapping	Capture high level ideas and concepts and trace them with down stream actions.
Business Rules	Identify and store business rules within your model while integrating them with downstream processes.
BPEL	Generate Business Process Execution Language (BPEL) from processes described using BPMN.
SPEM	A process engineering meta-model as well as conceptual framework, which provides the necessary concepts for

	modeling, documenting, presenting, managing, interchanging and enacting development methods and processes.
ArchiMate	Define your business capability with ArchiMate, an open-standard Enterprise Architecture language.
Eriksson-Penker Extensions	A framework for UML business processing model extensions.

Business Modeling/Interaction

Business Modeling diagrams and Business Interaction diagrams enable you to model both the structure and behavior of a business system. Business Modeling diagrams are based on a Class (UML Structural) diagram, whilst Business Interaction diagrams are based on a Sequence (UML Behavioral) diagram. Both diagram types have the same default Toolbox, which consists of a Business Modeling element page. The available elements include stereotyped Objects, and a stereotyped Actor (Business Actor), Use Case (Business Use Case) and Collaboration (Business Use Case Realization).

Example Diagram

[Example Business Modeling Diagram](#)

Example Business Modeling Diagram

This diagram shows the appearance of the elements dragged from the 'Business Modeling' pages of the Diagram Toolbox and dropped onto a Business Modeling diagram.
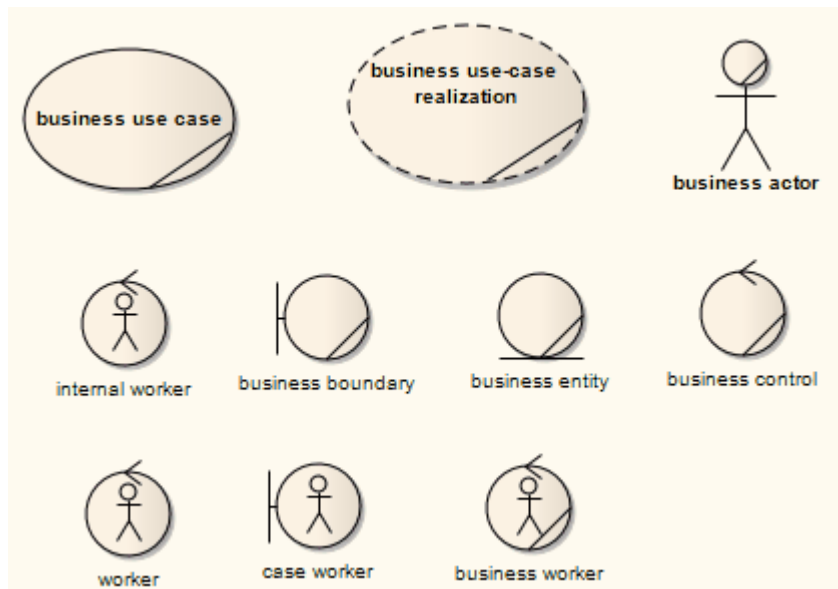
This diagram shows the appearance of the elements dragged from the 'Business Modeling' pages of the Diagram Toolbox and dropped onto a Business Interaction diagram.
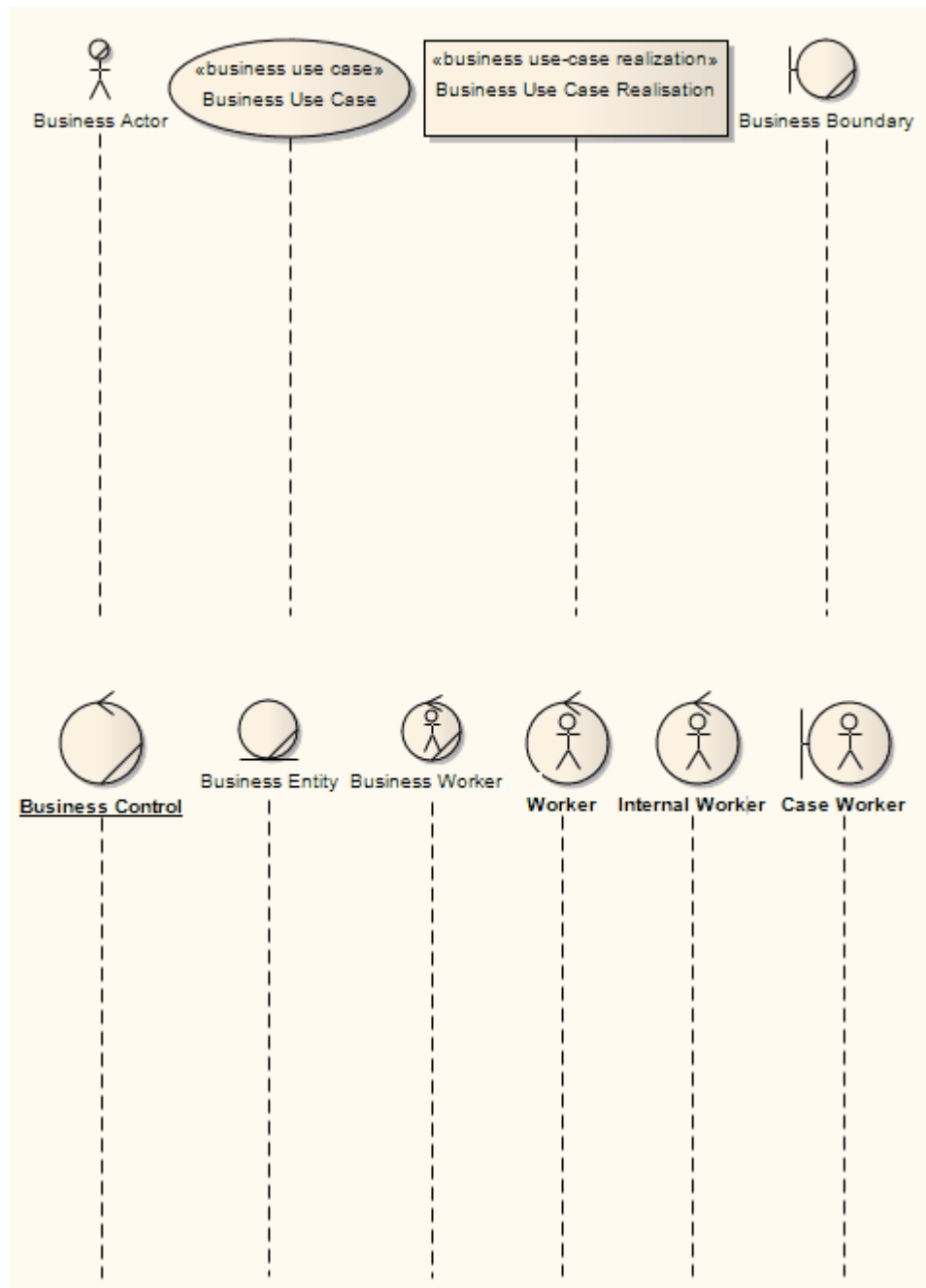
Risk Taxonomy

The UML Profile for Risk Taxonomy supports the Open Group Standard for Risk Taxonomy (OR-T), Version 2.0, which provides support for modeling risk scenarios and analyzing the risk conditions.

Risk Taxonomy Toolbox
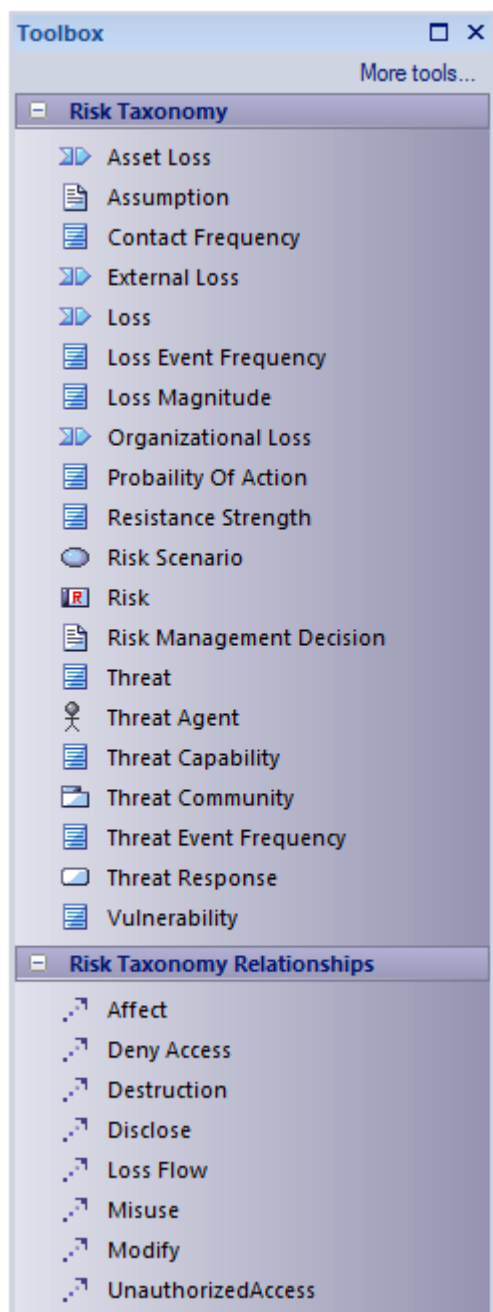
Elements

Icon	Description
Asset Loss	Defines the loss on an asset to the

	organization (in terms of the assets's value/liability and volume).
	Tagged Values
	• Liability/Value
	• Volume
	• Cost – The intrinsic value of the asset
	• Criticality – Very High (VH), High (H), Moderate (M), Low (L), Very Low (VL)
	• LossFactor
	• LossForm – Productivity, Response, Replacements, Fines and Judgements, Competitive Advantage, Reputation
	• LossType – Primary, Secondary
	• Sensitivity
	• Embarrassment/ Reputation - The information provides evidence of incompetent, criminal, or unethical management; note that this refers to reputation damage resulting from the nature of the information itself, as opposed to reputation damage that might result when a loss event takes place
	• Competitive Advantage - The information provides competitive advantage (such as key strategies or trade secrets); of the sensitivity

	categories, this is the only one where the sensitivity represents value - in all other cases, sensitivity represents liability • Legal Regulatory - The organization is bound by law to protect the information • General - Disclosure of the information results in some form of loss
Assumption	Captures assumptions made in risk analysis. Tagged Values • Rationale
Contact Frequency	The probable frequency, within a given timeframe, with which a threat agent will come into contact with an asset. Tagged Values • ConfidenceLevel - Commonly expressed as a percentage • MaximumLikelyValue • MinimumLikelyValue • Type – • Random - The threat agent 'stumbles upon' the asset during the course of unfocused or undirected activity • Regular - Contact occurs because of the

	regular actions of the threat agent • Intentional - The threat agent seeks out specific targets
External Loss	Captures external loss factors. Tagged Values • Factors – Event Detection, Legal and Regulatory, Competitors, Media, Other Stakeholders • Cost • Criticality – Very High (VH), High (H), Moderate (M), Low (L), Very Low (VL) • LossFactor • LossForm – Productivity, Response, Replacements, Fines and Judgements, Competitive Advantage, Reputation • LossType – Primary, Secondary • Sensitivity – • Embarrassment/ Reputation - The information provides evidence of incompetent, criminal, or unethical management; note that this refers to reputation damage resulting from the nature of the information itself, as opposed to reputation damage that can result when a loss event takes place • Competitive Advantage - The

	information provides competitive advantage (such as key strategies or trade secrets); of the sensitivity categories, this is the only one where the sensitivity represents value - in all other cases, sensitivity represents liability • Legal Regulatory - The organization is bound by law to protect the information • General - Disclosure of the information results in some form of loss
Loss	Captures loss that a threat can result in. Tagged Values • Cost • Criticality – Very High (VH), High (H), Moderate (M), Low (L), Very Low (VL) • LossFactor • LossForm – Productivity, Response, Replacements, Fines and Judgements, Competitive Advantage, Reputation • LossType – Primary, Secondary • Sensitivity • Embarrassment/ Reputation - The information provides evidence of incompetent, criminal, or unethical management; note that this refers to

	reputation damage resulting from the nature of the information itself, as opposed to reputation damage that might result when a loss event takes place • Competitive Advantage - The information provides competitive advantage (such as key strategies or trade secrets); of the sensitivity categories, this is the only one where the sensitivity represents value - in all other cases sensitivity represents liability • Legal Regulatory - The organization is bound by law to protect the information • General - Disclosure of the information results in some form of loss
Loss Event Frequency	The probable frequency, within a given timeframe, that a threat agent will inflict harm upon an asset. Tagged Values • LossType • ProbableFrequency - Probability is always is based on a timeframe (event X is 10% likely to occur over the next Y) • Rating - Very High (VH), High (H),

	Moderate (M), Low (L), Very Low (VL) • TimePeriod
Loss Magnitude	The probable magnitude of loss resulting from a loss event. Tagged Values • LossType • Rating - Severe (SV), High (H), Significant (Sg), Moderate (M), Low (L), VeryLow (VL)
Organization al Loss	Captures the loss to the organization. Tagged Values • Factors – Timing, DueDiligence, Response, Detection • Cost – The intrinsic value of the asset • Criticality – Very High (VH), High (H), Moderate (M), Low (L), Very Low (VL) • LossFactor • LossForm • Productivity - The reduction in an organization's ability to generate its primary value proposition • Response - Expenses associated with managing a loss event (such as internal

| | or external person-hours, logistical expenses, legal defense and public relations expenses)

- Replacement - The capital expense associated with replacing lost or damaged assets

- Fines and Judgements - Legal or regulatory actions levied against an organization

- Competitive Advantage - Losses associated with diminished competitive position

- Reputation - Losses associated with an external stakeholder's perception that an organization's value proposition is diminished

- LossType – Primary, Secondary

- Sensitivity

- Embarrassment/ Reputation - The information provides evidence of incompetent, criminal, or unethical management; note that this refers to reputation damage resulting from the nature of the information itself, as opposed to reputation damage that might result when a loss event takes place

- Competitive Advantage - The |

	information provides competitive advantage (such as key strategies or trade secrets); of the sensitivity categories, this is the only one where the sensitivity represents value - in all other cases, sensitivity represents liability • Legal Regulatory - The organization is bound by law to protect the information • General - Disclosure of the information results in some form of loss
Probability of Action	The probability that a threat agent will act against an asset once contact occurs. Tagged Values • ConfidenceLevel - Commonly expressed as a percentage • LevelOfEffort - Very High (VH), High (H), Moderate (M), Low (L), Very Low (VL) • MaximumLikelyValue - T he threat agent's maximum value proposition from performing the act • MinimumLikelyValue - The threat agent's minimum value proposition from performing the act • MostLikelyValue - The threat agent's perceived value proposition from

	performing the act
	• RiskOfDetection/Capture - Very High (VH), High (H), Moderate (M), Low (L), Very Low (VL)
Resistance Strength	The strength of a control as compared to a baseline measure of force.
	Tagged Values
	• ConfidenceLevel - Commonly expressed as a percentage
	• LevelOfEffort - Very High (VH), High (H), Moderate (M), Low (L), Very Low (VL)
	• MaximumLikelyValue
	• MinimumLikelyValue
	• MostLikelyValue
	• Rating - Very High (VH), High (H), Moderate (M), Low (L), Very Low (VL)
Risk Scenario	Defines the environment or situation that involves risk.
Risk	Defines an estimate of the probable frequency and magnitude of future loss
	Tagged Values
	• Rating - Very High (VH), High (H), Moderate (M), Low (L), Very Low

	(VL)
Risk Management Decision	The decision to mitigate the risk.
Threat	Defines the harm to an asset that a risk poses. Tagged Values • ThreatScenario – Malicious, Error, Failure, Natural • ThreatType – External, Internal
Threat Agent	The source that could inflict harm upon an asset. Tagged Values • AccessMethod • DesiredVisibility • Motive • Objective • PersonalRiskTolerance • Resources • SkillRating • Sponsorship • ThreatType – External, Internal
Threat	The probable level of force that a threat

Capability	agent is capable of applying against an asset. **Tagged Values** • ConfidenceLevel - Commonly expressed as a percentage • MaximumLikelyValue - The maximum level of skills an attacker might have • MinimumLikelyValue - The minimum level of skills that an attacker might have • MostLikelyValue - The skill level of the most likely attacker • Rating - Very High (VH), High (H), Moderate (M), Low (L), Very Low (VL)
Threat Community	People associated with the conditions surrounding the asset at risk. **Tagged Values** • ThreatType – External, Internal
Threat Event Frequency	The probable frequency, within a given timeframe, at which a threat agent will act against an asset. **Tagged Values** • Rating - Very High (VH), High (H), Moderate (M), Low (L), Very Low (VL)

Threat Response	The action associated with managing a loss event. Tagged Values • ResponsePercentage • ResponseType – • Containment - An organization's ability to limit the breadth and depth of an event • Remediation - An organization's ability to remove the threat agent • Recovery - The ability to bring things back to normal
Vulnerability	The probability that a threat event will become a loss event. Tagged Values • Rating - Very High (VH), High (H), Moderate (M), Low (L), Very Low (VL)

Relationships

Icon	Description

Affect	Represents an impact that a threat has on an asset.
Deny Access	Represents a relationship type of a threat to an asset.
Destruction	Represents loss on an asset, such as destruction or theft of a non-data asset.
Disclose	Represents an agent illicitly disclosing sensitive information.
Loss Flow	Represents mapping between losses.
Misuse	Represents unauthorized use of assets (such as identity theft, or setting up an illegal distribution service on a compromised server).
Modify	Represents unauthorized changes to am asset.
Unauthorized Access	Represents a simple unauthorized access relationship on an asset.

Business Motivation Model

This text is derived from the Business Motivation Model Specification document (OMG document, BMM 1.3, dtc-14-11-06, Section 1.1, pg 1).

"The Business Motivation Model provides a scheme or structure for developing, communicating, and managing business plans in an organized manner. Specifically, the Business Motivation Model does all of the following:

- Identifies factors that motivate the establishing of business plans.

- Identifies and defines the elements of business plans.

- Indicates how all these factors and elements inter-relate

Among these elements are those that provide governance for and guidance to the business — Business Policies and Business Rules.

The structure of the BMM provides the basis for logical design of easily-constructed tools for storing, cross-referencing, and reporting the elements of business plans for business enterprises."

The Business Motivation Model (BMM) is integrated with Enterprise Architect through the MDG Technology for the Business Motivation Model, provided with the Enterprise Architect installer.

This feature is available in the Corporate, Unified and Ultimate editions of Enterprise Architect, from Release 15.0.

Access

Click on the 'Perspective' icon (top right corner of the display) and select 'Business Modeling > Business Motivation'; the Model Wizard displays.

Select either the:

- 'Model Patterns' tab and one of the Patterns to add a template BMM model to the selected Package, or
- The 'Diagram' tab and select 'BMM > BMM Diagram' to add an empty Business Motivation Model diagram to the selected Package; the 'BMM' page displays in the Diagram Toolbox

Modeling with the Business Motivation Model

Developing models from the Business Motivation Model is not hard; there are many types of element and connector, providing a broad and versatile platform from which to create diagrams and models. The Business Motivation Model elements are simple in structure, consisting principally of just the name or ID and text, making it easy to quickly build up a structure.

Toolbox

Search

⊿ BMM - Ends
- ☑ End
- ☑ Desired Result
- ☑ Goal
- ☑ Objective
- ☑ Vision

⊿ BMM - Means
- ☑ Means
- ☑ Mission
- ☑ Course of Action
- ☑ Strategy
- ☑ Tactic
- ☑ Directive
- ☑ Business Rule
- ☑ Business Policy

⊿ BMM - Influencers
- ☑ Influencer
- ☑ Influencing Organization
- ☑ Organization Unit
- ☑ Regulation
- ☑ Business Process

⊿ BMM - Assessments
- ☑ Assessment
- ☑ Potential Impact

⊿ BMM - Categories
- 📄 Desired Result Category
- 📄 Influencer Category
- 📄 Organization Category
- 📄 Assessment Category

⊿ BMM - Asset and Liability
- 📄 Asset
- 📄 Fixed Asset
- 📄 Offering
- 📄 Liability
- 📄 Resource

Toolbox

Search

⊿ BMM - Relationships
- ↗ Acts As
- ↗ Affects Achievement Of
- ↗ Affects Employment Of
- ↗ Amplifies
- ↗ Categorizes
- ↗ Channels Efforts Towards
- ↗ Claims
- ↗ Defines
- ↗ Delivers
- ↗ Deploys
- ↗ Determines
- ↗ Discharges
- ↗ Effects Enforcement Level Of
- ↗ Enables
- ↗ Establishes
- ↗ Governs
- ↗ Governs Use Of
- ↗ Guides
- ↗ Identifies
- ↗ Implements
- ↗ Includes
- ↗ Is Formulated Based On
- ↗ Is Judgment Of
- ↗ Is Responsible For
- ↗ Is Source Of
- ↗ Makes
- ↗ Makes Operative
- ↗ Manages
- ↗ Provides
- ↗ Provides Impetus For
- ↗ Quantifies
- ↗ Realizes
- ↗ Recognizes
- ↗ Requires
- ↗ Supports Achievement Of
- ↗ Uses

Facility	Description
BMM in	You can develop Business Motivation

Enterprise Architect	Model diagrams in Enterprise Architect quickly and easily, using the MDG Technology for the Business Motivation Model. The BMM facilities are provided in the form of: • A BMM diagram type, created through the 'New Diagram' dialog ('Business Modeling > BMM') • A 'BMM' page in the Diagram Toolbox, providing BMM elements (stereotyped UML elements) • BMM element and relationship entries in the 'Toolbox Shortcut Menu' and Quick Linker

Example Diagram

You can re-create this diagram from the 'BMM Example - Pizza Company' Pattern on the 'Business Motivation Perspective' page of the 'Model Patterns' tab of the Model Wizard.

Pizza Company. BMM Diagram

«Vision»
Be the city's favorite pizza place.

«Mission»
Provide pizza to customers city-wide.

«MakesOperative»

«Amplifies»

«Goal»
To deliver pizzas in an expedient amount of time.

**«Strategy»**
Deliver pizzas to the location of the customer's choice.

«ChannelsEffortsTowards»

«Quantifies»

«Implements»

«Objective»
By January 1, 2007, 95% on-time pizza delivery.

«Tactic»
Hire drivers with their own vehicles to deliver pizzas.

«ChannelsEffortsTowards»

«SupportsAchievementOf»

«BusinessRule»
Pizzas may not be delivered beyond a radius of 30 miles.

«BusinessPo...»
Safety in the kitchen, and in the streets, comes first.

«BusinessR...»
Pizzas must be delivered within one hour.

«BusinessRule»
A driver who is convicted of one moving traffic violation while driving for the company will be counseled by a manager concerning safe driving.

tags
EnforcementLevel = pre-authorized override

«Influencer»
Influencer 1

notes
"Depreciation starts when you first use the property in your business or for the production of income. It ends when you take the property out of service, deduct all your depreciable cost or other basis, or no longer use the property in your business or for the production of income."

IRS 1999 1040 Forms and Instructions, 1999 Instructions for Schedule C, Profit and Loss from a Business Part II. Expensis, p. C-3

«InfluencerCateg...»
Regulation

«Categorizes»

Eriksson-Penker Extensions

Eriksson-Penker extensions (developed by H. E. Eriksson and M. Penker) provide a framework for UML business processing model extensions, to which an enterprise architect can add stereotypes and properties appropriate to their business.

Access

On the Diagram Toolbox, click on to display the 'Find Toolbox Item' dialog and specify 'Eriksson-Penker Extensions'.

Ribbon	Design > Diagram > Toolbox
Keyboard Shortcuts	Ctrl+Shift+3

Toolbox Page
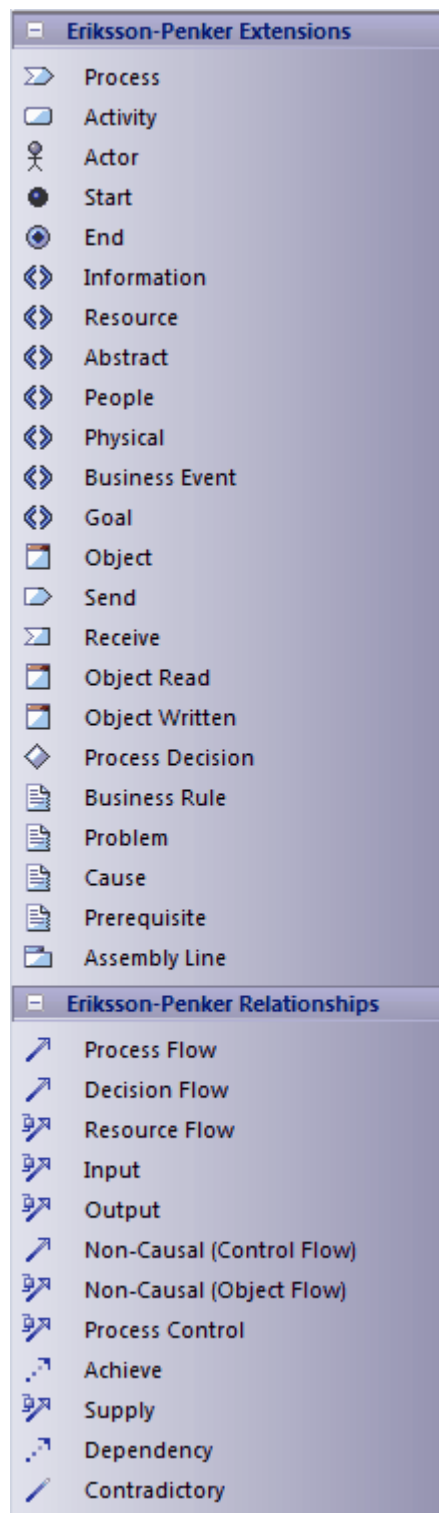
Eriksson-Penker in Enterprise Architect

Enterprise Architect provides - through the integration of MDG Technologies with the installer - two well-respected and proven UML extensions that further enhance the capture of business activities, processes, objects and information flows.

One of these is Business Process Model and Notation (BPMN).

The other is the Eriksson-Penker profile which, through a set of stereotypes, provides a unique and powerful means of visualizing and communicating business processes and the necessary flow of information within an organization.

The Eriksson-Penker extensions are provided in the form of:

- An Eriksson-Penker diagram type, accessed through the 'New Diagram' dialog
- An Eriksson-Penker page in the Toolbox
- Eriksson-Penker element and relationship entries in the 'Toolbox Shortcut' menu and Quick Linker

Example

This is an example of a simple Eriksson-Penker diagram:
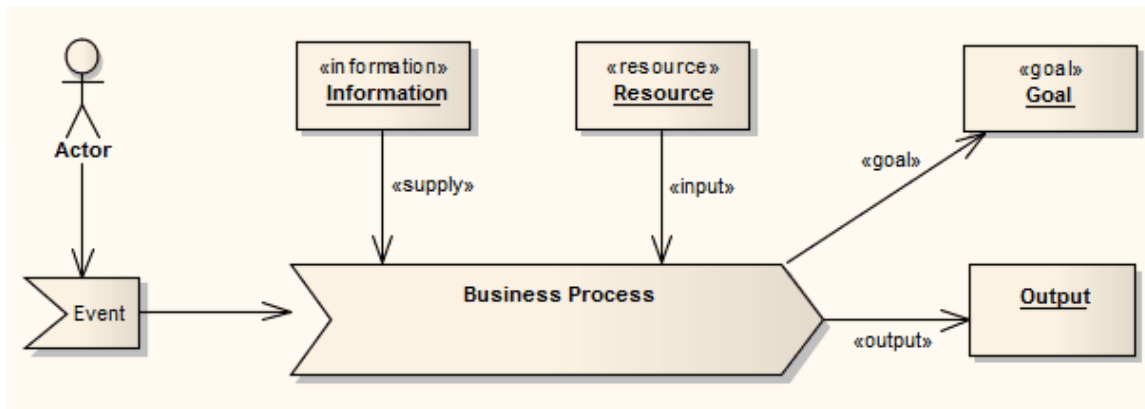
Business Models

Modeling the Business Process

Modeling the business process is an essential part of any software development process. It enables the analyst to capture the broad outline and procedures that govern what it is a business does. This analysis model provides an overview of where the proposed software system being considered fits into the organizational structure and daily activities. It can also provide the justification for building the system by capturing the current manual and automated procedures that are to be rolled up into a new system, and the associated cost benefit.

As an early model of business activity, it enables the analyst to capture the significant events, inputs, resources and outputs associated with business process. By connecting later design elements (such as Use Cases) back to the business process model through Implementation connectors, it is possible to build up a fully traceable model from the broad process outlines to the functional requirements and eventually to the software artifacts actually being constructed.

As the Business Process Model typically has a broader and more inclusive range than just the software system being considered, it also enables the analyst to clearly map what is in the scope of the proposed system and what is to be implemented in other ways (such as a manual process).

Example

The example diagram demonstrates the kind of model that can be built up to represent a business process. In this model, the goal of the business process is to take customer orders and to ship those orders out. A user starts the process with an inquiry, which leads to the involvement of the Book Catalogue, Shopping Cart, on-line pages and warehouse inventory. The output of significance to the business is a customer order.
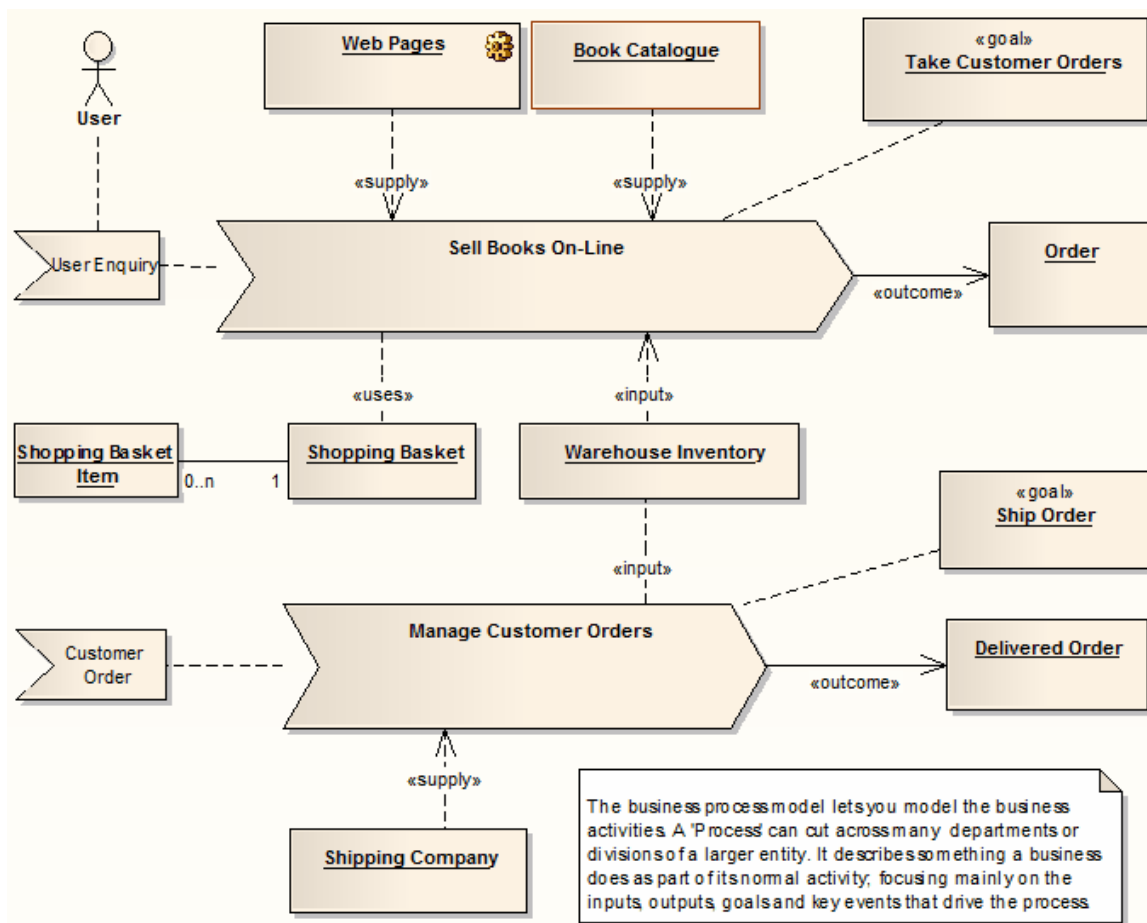
The second half of the process model is to respond to a customer order and ship the required items. The second

process involves the warehouse inventory and shipping company, and completes when an order is delivered to the customer.

Analysis Models

This section discusses the development of Analysis models to construct business processes. It describes:

- [Process Modeling Notation](#)
- [Inputs, Resources and Information](#)
- [Events](#)
- [Outputs](#)
- [Goals](#)
- [A Complete Business Process](#)

Process Modeling Notation

A business process is a collection of activities designed to produce a specific output for a particular customer or market. It implies a strong emphasis on how the work is done within an organization, in contrast to a product's focus on what is produced. A process is thus a specific ordering of work activities across time and place, with a beginning, an end, and clearly defined inputs and outputs: a structure for action. The notation used to depict a business process is illustrated here.
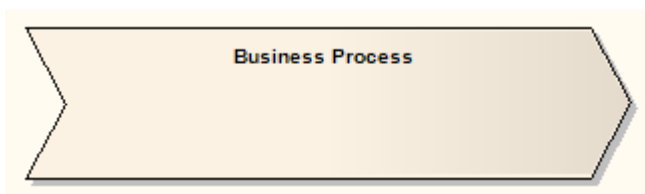
The business process:

- Can affect more than one organizational unit
- Can have a horizontal organizational impact
- Creates value of some kind for the customer; customers can be internal or external

A business process model typically defines these elements:

- The goal or reason for the process
- Specific inputs
- Specific outputs
- Resources consumed
- Activities that are performed in some order, and
- Events that drive the process

The process notation implies a flow of activities from left to right. Typically an Event element is placed to the left of the process and the output to the right. To specifically notate the internal activities, Activity elements can be placed inside the Process element.

The BPMN File

One popular notation and approach to business modeling is the Business Process Model and Notation (BPMN). This notation is specifically targeted at the business modeling community and has a relatively direct mapping to UML through a BPMN Profile. Sparx Systems provides a built-in profile for BPMN in Enterprise Architect.

Inputs, Resources and Information

Business processes use information to tailor or complete their activities. Information, unlike resources, is not consumed in the process; rather it is used as part of the transformation process. Information can come from external sources, from customers, from internal organizational units and could even be the product of other processes.

A resource is an input to a business process and, unlike information, is typically consumed during the processing. For example, as each daily train service is run and actuals recorded, the service resource is 'used up' as far as the process of recording actual train times is concerned.

The notation to illustrate information and resources is shown here.
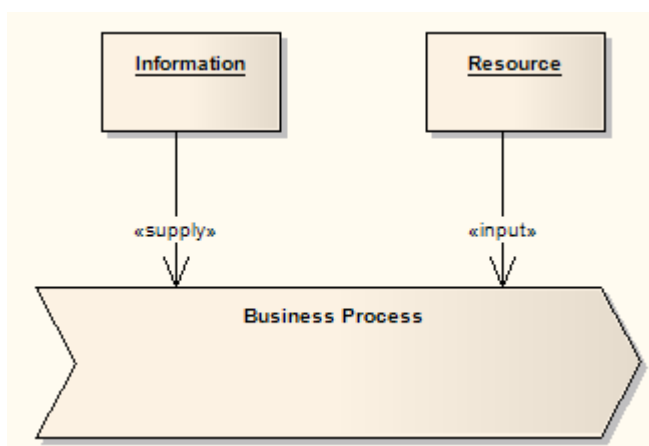
A Supply connector indicates that the information or object linked to the process is not used up in the processing phase. For example, order templates can be used over and over to provide new orders of a certain style; the templates are not altered or exhausted as part of this activity.

An Input connector indicates that the attached object or

resource is consumed in the processing procedure. As an example, as customer orders are processed they are completed and signed off, and typically are used only once per unique resource (order).

Events

An event is the receipt of some object, a time or date reached, a notification or some other trigger that initiates the business process. The event might be consumed and transformed (for example a customer order) or simply act as a catalyst (for example, nightly batch job).
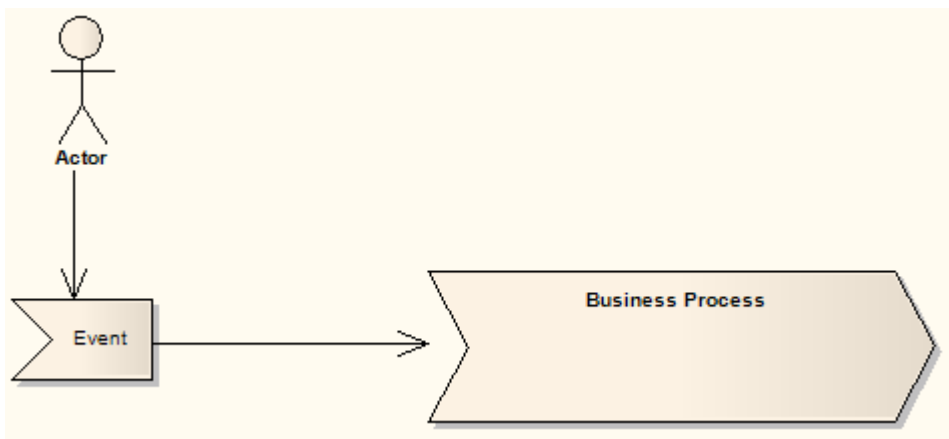
Outputs

A business process typically produces one or more outputs of value to the business, either for internal use or to satisfy external requirements. An output might be a physical object (such as a report or invoice), a transformation of raw resources into a new arrangement (a daily schedule or roster) or an overall business result such as completing a customer order.

An output of one business process might feed into another process, either as a requested item or a trigger to initiate new activities.
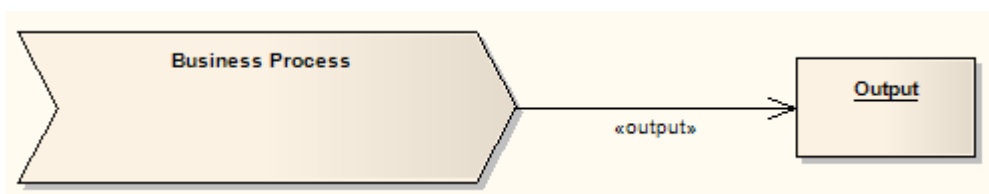
An Output connector indicates that the business process produces some object (either physical or logical) that is of value to the organization, either as an externally visible item or as an internal product (possibly feeding another process).

Goals

A business process has some well defined goal. This is the reason the organization does this work, and should be defined in terms of the benefits this process has for the organization as a whole and in satisfying the business requirements.
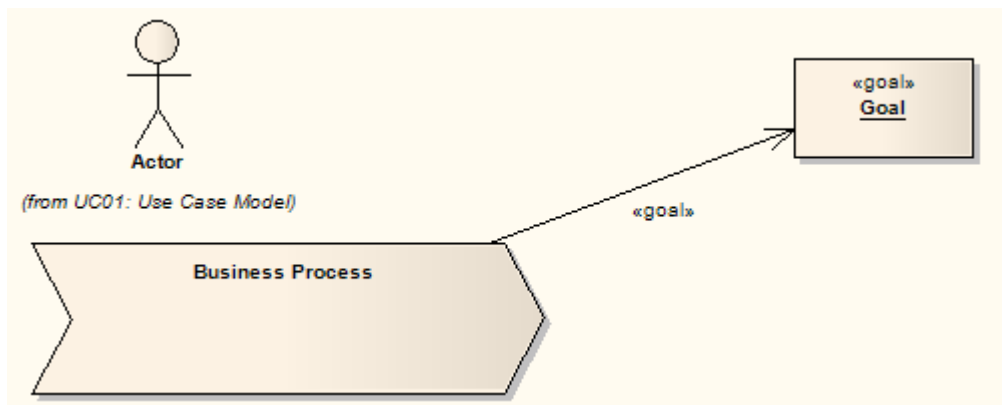
A Goal connector indicates that the attached object to the business process describes the goal of the process. A goal is the business justification for performing the activity.

A Complete Business Process

This diagram illustrates how the various model elements can be grouped together to produce a coherent picture of a named business process. Included are the inputs, outputs, events, goals and other resources that are of significance.
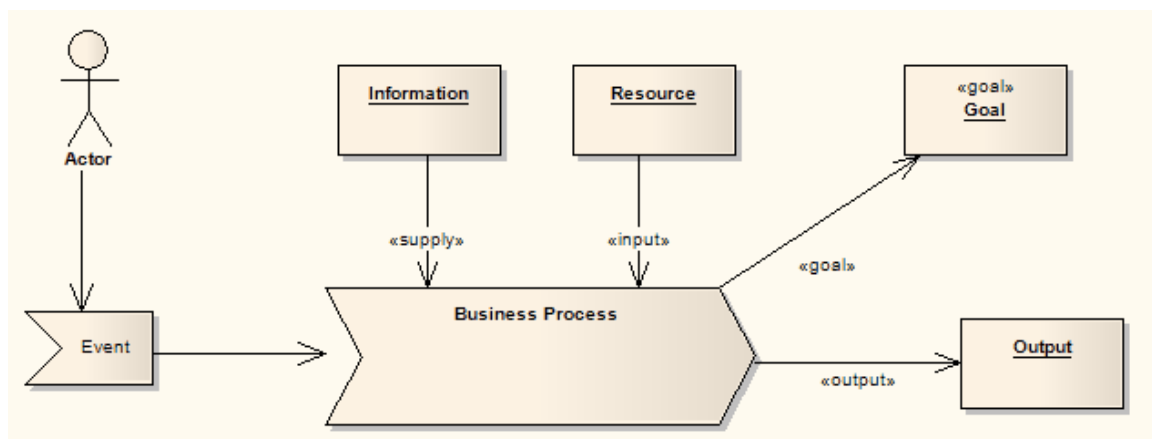
SPEM

The MDG Technology for SPEM 2.0 is one of the modeling tools integrated with Enterprise Architect.

"The Software and Systems Process Engineering Metamodel (SPEM) is a process engineering meta-model as well as conceptual framework, which can provide the necessary concepts for modeling, documenting, presenting, managing, interchanging, and enacting development methods and processes. An implementation of this meta-model would be targeted at process engineers, project leads, project and program managers who are responsible for maintaining and implementing processes for their development organizations or individual projects."

(Quoted from the Object Management Group (OMG) Software & Systems Process Engineering Meta-Model Specification (Version 2.0, April 01 2008))

SPEM is a Profile of UML, which uses UML as a notation and takes an object-oriented approach. To accommodate UML 2, the SPEM specification was upgraded to 2.0 in April 2008. SPEM 2.0 focuses on providing the additional information structures that you require for processes modeled with UML 2 Activities or BPMN/BPDM, to describe an actual development process.

Access

Show the Diagram Toolbox using any of the methods outlined here.

On the Diagram Toolbox, click on 🔎 to display the 'Find Toolbox Item' dialog and specify 'SPEM'.

Ribbon	Design > Diagram > Toolbox
Keyboard Shortcuts	Ctrl+Shift+3
Other	Click the » icon on the Diagram caption bar to display the Diagram Toolbox

SPEM Integration

Use	Discussion
SPEM in Enterprise Architect	Developing SPEM diagrams is quick and simple, using the MDG Technology for SPEM 2.0. The SPEM facilities are provided in the form of: • A SPEM diagram type, accessed through the 'New Diagram' dialog • A set of SPEM pages in the Diagram Toolbox, providing SPEM elements

| | (stereotyped UML elements) |
| | • SPEM element and relationship entries in the 'Toolbox Shortcut Menu' and Quick Linker |

SPEM Toolbox Pages

The SPEM pages of the Diagram Toolbox provide a wide range of specialized elements and connectors for modeling development processes on SPEM diagrams. To create elements and relationships on a SPEM diagram, you can drag the appropriate icons from the Toolbox pages.

Access

Show the Diagram Toolbox using any of the methods outlined here.

On the Diagram Toolbox, click on to display the 'Find Toolbox Item' dialog and specify 'SPEM'.

Ribbon	Design > Diagram > Toolbox : to display the 'Find Toolbox Item' dialog and specify 'SPEM'
Keyboard Shortcuts	Ctrl+Shift+3 : ≡ | SPEM
Other	Click the » icon on the Diagram caption bar to display the Diagram Toolbox

Toolbox Pages

Base Plug-in

- Phase
- Iteration
- Process
- DeliveryProcess
- Process Pattern
- Process Planning Template
- Artifact
- Deliverable
- Outcome

Method Content

- Category
- Guidance
- Metric
- Role Definition
- Step
- Task Definition
- Tool Definition
- Work Product Definition

Package

- Method Content Package
- Method Library
- Method Plugin
- Process Package
- Process Component
- Work Product Port
- Expose Interface

Process

- Activity
- Composite Role
- Milestone
- Role Use
- Task Use
- Team Profile
- Work Product Use

Relationship

- Association
- Composition
- Control Flow
- Object Flow
- Specialization
- Nesting
- Package Import

Each element and connector provided in the SPEM Toolbox pages is described in this table.

Base Plug-in Toolbox Page

Object	Action
Phase	Create a predefined special Activity representing a significant period in a project.
Iteration	Group a set of nested Activities that are repeated more than once. Typically, Iteration is an Activity for which the default value of the isRepeatable attribute is True.
Process	Represents a special Activity that describes a structure for particular types of development project, or parts of them.
Delivery Process	Represent a special Process describing a complete and integrated approach for implementing a specific project type.
Process Pattern	Represent a special Process to describe a reusable cluster of Activities in a general

	process area that provides a consistent development approach to common problems.
Process Planning Template	Represent a special Process that is prepared for instantiation by a project planning tool.
Artifact	Represent a Work Product Definition that provides a description and definition for tangible work product types.
Deliverable	Represent a Work Product Definition that provides a description and definition for packaging other Work Products, and that can be delivered to an internal or external party.
Outcome	Represent a Work Product Definition that provides a description and definition for non-tangible work products.

Method Content Toolbox Page

Object	Action

Category	Categorize content based on the user's criteria.
Guidance	Identify reference items such as Guidelines, Templates, Checklists, Tool Mentors, Estimates, Supporting Materials, Reports and Concepts.
Metric	Define a standard measurement for instances of Method Content elements.
Role Definition	Define a set of related skills, competencies, and responsibilities.
Step	Represent parts or subunits of a Task Definition.
Task Definition	Describe an assignable unit of work. Every Task Definition is assigned to specific Role Definitions. A Task is associated with input and output Work Products.
Tool Definition	Describe the tools that are recommended or necessary for completing a specific Task.
Work Product	Define any forms of document, report or outcome that are consumed, produced or

Definition	modified by Tasks.

Package Toolbox Page

Object	Action
Method Content Package	Create a physical container to organize the Method Content elements.
Method Library	Create an overall physical container for all SPEM 2.0 elements.
Method Plug-in	Create a physical container for Method Content Package and Process Packages. It can be used stand-alone as well as extended to many other Method Plug-ins.
Process Package	Create a physical container that contains different kinds of Process element.
Process Component	Offers the choice of creating a Process Component Package - a special type of Process Package that provides the mechanism of encapsulation - or a Process Component element.

| Work Product Port	Defines the Work Products input and output for a Process Component. A Work Product Port identifies one type of Work Product for one Process Component, and defines whether the Work Product type is to be required (input) or supplied (output) by the Process Component, and whether this input or output is optional or not.
Expose Interface	Represents a required or supplied interface to identify the fact that the element provides or requires an interface. A small dialog displays on which you type or select the name of the interface and whether it is provided or required.

Process Toolbox Page

Object	Action
Activity	Define basic units of work within a Process as well as the Process itself.
Composite Role	Represent an aggregation of Role Definition references for an Activity.

| Milestone	Represent any significant events in a development project.
Process	Create a special Activity that describes a structure for particular types of development project.
Role Use	Represent a Role Definition in the context of one specific Activity.
Task Use	Represent a Task Definition in the context of one specific Activity.
Team Profile	Define a nested hierarchy of teams and team members.
Work Product Use	Represent a Work Product Definition in the context of one specific Activity.

Example SPEM Diagram

You can model the development processes underpinning software and business process modeling using SPEM diagrams, and the wide range of specialized elements and connectors provided in the 'SPEM' pages of the Diagram Toolbox.

Access

Ribbon	Design > Diagram > Add > Type : Management > SPEM, Select From = 'SPEM', Diagram Types = 'SPEM_DP'
Context Menu	Browser window | Right-click on Package | Add Diagram : Type = Management | SPEM : Select From = 'SPEM', Diagram Types = 'SPEM_DP'
Keyboard Shortcuts	
Other	Browser window header bar : ☰▾ | Add Diagram : Type = Management | SPEM : Select From = 'SPEM', Diagram Types = 'SPEM_DP'

Example Diagram
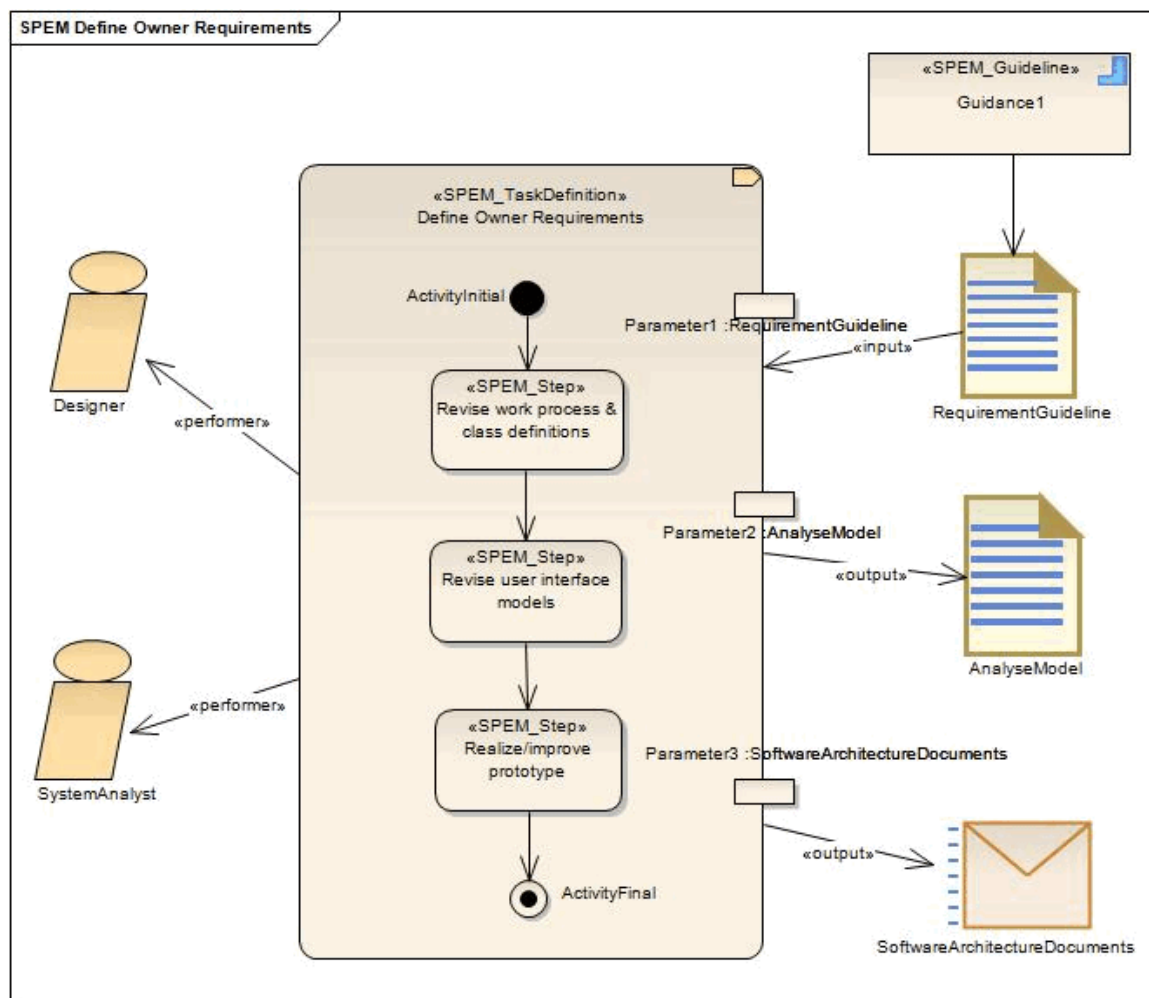
SPEM Element appearance

Every SPEM stereotype can be presented in one of two ways:

- Iconic presentation, where the shape is the object's icon

(as shown for Designer, which has the Role Definition icon shape), or

- Textual presentation, where the shape is a rectangular outline with the object's icon as a decoration in the top right corner (as shown for SPEM_TaskDefinition)

To set the presentation format for an element, use the iconstyle Tagged Value. To use:

- Iconic presentation, set the iconstyle Tagged Value to True

- Textual presentation, set the Tagged Value to False

MDG Technology for AML

The MDG Technology for the Archetype Modeling Language (AML) integrates the AML modeling tool with Enterprise Architect.

This text is derived from the 'Object Management Group (OMG) Archetype Modeling Language (AML) Specification' (Version 1.0, May 2015).

"The AML defines a standard means for modeling Archetype Models (AMs) to support the representation of Clinical Information Modeling Initiative (CIMI) artifacts using modeling profiles as defined in the UML. Archetype Models are Platform Independent Models (PIMs) and are developed as a set of constraints on a specific Reference Model (RM)."

The AML is specified by three UML profiles collectively meeting the requirements of archetype modeling:

- Reference Model
- Constraint Model
- Terminology Binding Model

This image illustrates the selection of the 'AML' diagram type in the 'New Diagram' dialog.
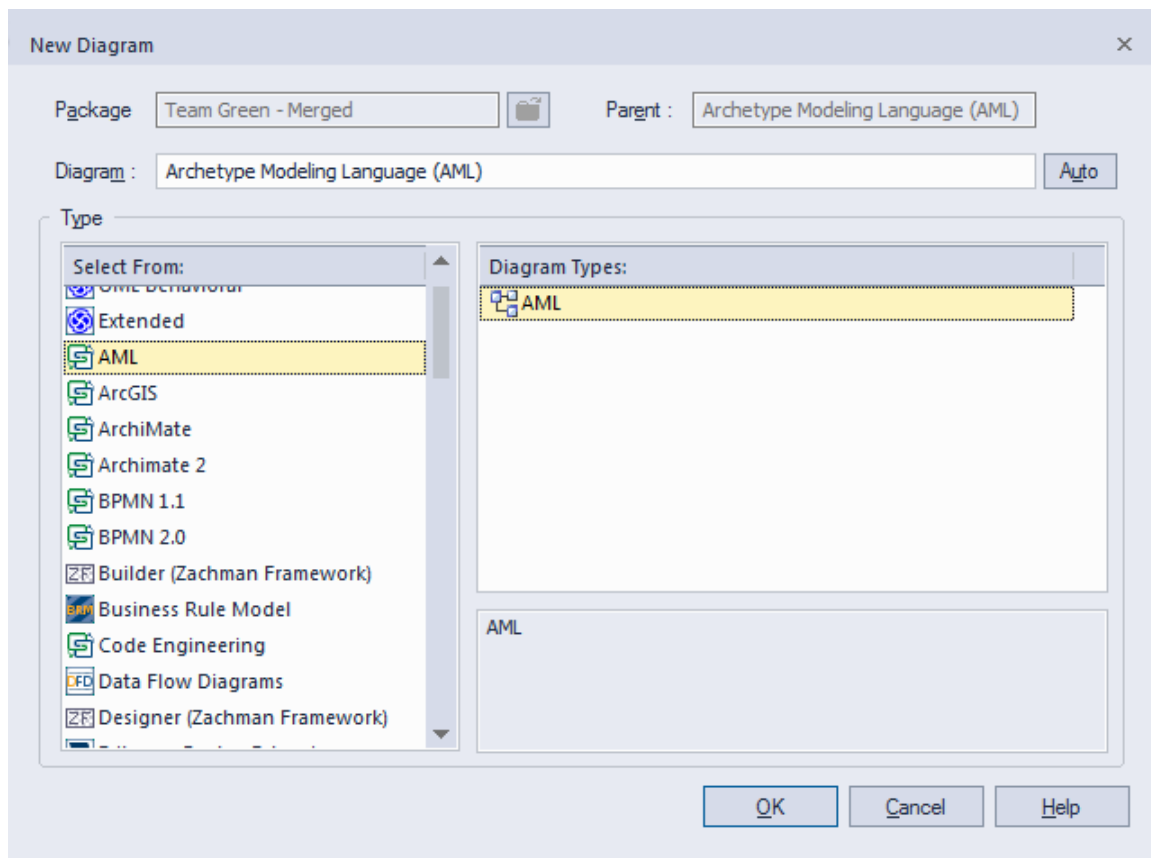
Access

On the Diagram Toolbox, click on to display the 'Find Toolbox Item' dialog and specify 'Archetype Modeling Language'.

Ribbon	Design > Diagram > Toolbox
Keyboard Shortcuts	Ctrl+Shift+3

AML Integration

| Facility	Description
AML in Enterprise Architect	Developing AML diagrams is quick and simple, using the MDG Technology for Archetype Modeling Language. The AML facilities are provided in the form of: - An AML diagram type, created through the 'New Diagram' dialog - An 'Archetype Modeling Language' page in the Diagram Toolbox, providing AML elements (stereotyped UML elements) - AML element and relationship entries in the 'Toolbox Shortcut Menu' and Quick Linker

MDG Technology for CMMN

The MDG Technology for the Case Management Model & Notation (CMMN) allows you to create CMMN models with Enterprise Architect.

"A Case is a proceeding that involves actions taken regarding a subject in a particular situation to achieve a desired outcome. Traditional examples come from the legal and medical worlds, where a legal Case involves the application of the law to a subject in a certain fact situation, and a medical Case involves the care of a patient in the context of a medical history and current medical problems."

(Quoted from the Object Management Group (OMG) Case Management Model and Notation Specification (Version 1.0, May 2014))

This image shows a CMMN diagram created with Enterprise Architect:
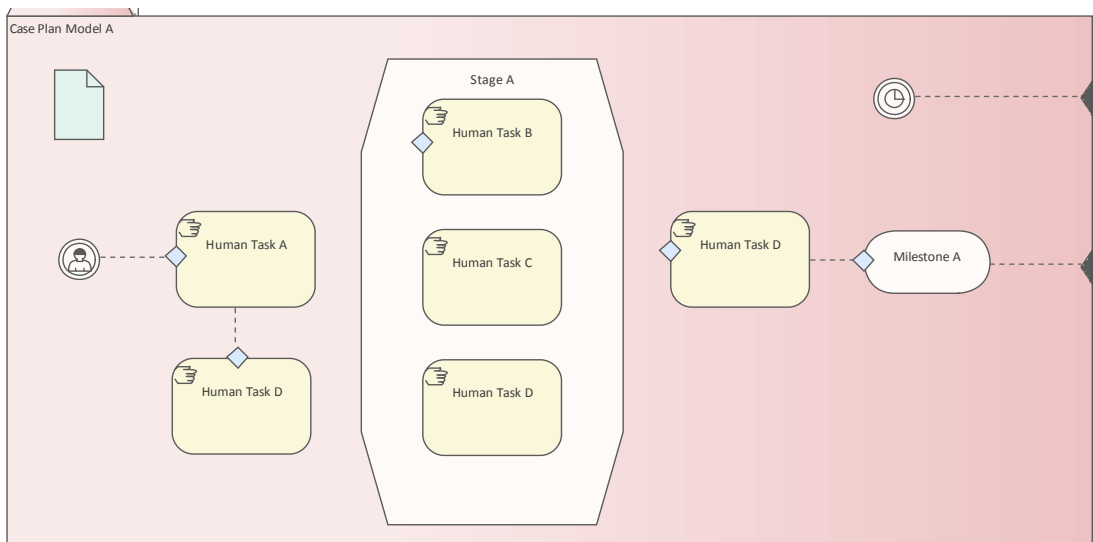
Access

Click on the 'Perspective' icon (top right corner of the display) and select Perspective > Analysis > CMMN; ; the Model Wizard displays.

Select either the:

- 'Model Patterns' tab and one of the Patterns to add a template CMMN model to the selected Package, or
- The 'Diagram' tab and select 'CMMN > CMMN' to add an empty CMMN diagram to the selected Package; the CMMN page displays in the Diagram Toolbox

Ribbon	Design > Diagram > Toolbox

CMMN Modeling

Facility	Description
CMMN in Enterprise Architect	Developing CMMN diagrams is quick and simple, using the MDG Technology for CMMN. The CMMN facilities are provided in the form of: • A CMMN diagram type, created through the 'New Diagram' dialog • A 'CMMN' page in the Diagram

| | Toolbox, providing CMMN elements (stereotyped UML elements) |
| | • CMMN element and relationship entries in the 'Toolbox Shortcut Menu' and Quick Linker |

MDG Technology for VDML

The MDG Technology for the Value Delivery Modeling Language (VDML) allows you to create VDML models with Enterprise Architect.

This definition of VDML is derived from the Object Mangement Group (OMG) Value Delivery Modeling Language (VDML) Specification (version 1.0, October 2015):

"The purpose of VDML is to provide a standard modeling language for analysis and design of the operation of an enterprise with particular focus on the creation and exchange of value. It provides an abstraction of the operation of an enterprise that is appropriate for business executives, along with representation of supporting detail for business analysts to link strategy and business models to the activities, roles, and capabilities that run the enterprise. The target users are business people-executives, business architects, analysts and managers. Information systems analysts and designers may use VDML models as specifications for the design of supporting information systems."
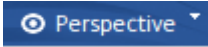
This image shows an example of one of the VDML diagrams that can be created with Enterprise Architect:
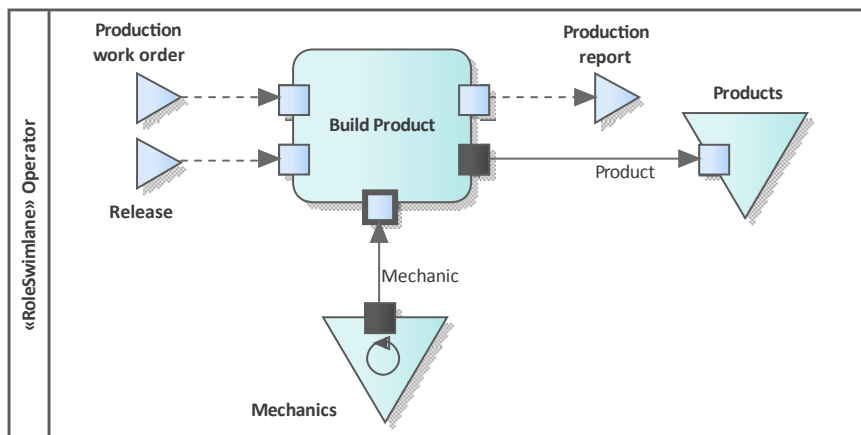
This feature is available in the Unified and Ultimate editions of Enterprise Architect, from Release 15.0.

Access

For the selected Package, open the Model Wizard, select the required Viewpoint and Pattern, then click on the Create Patterns button. The model pattern is created in the selected Package. When you open the diagram created from the Pattern, the Diagram Toolbox automatically displays the 'VDML' Toolbox page corresponding to the diagram type you have opened.

Ribbon	Design > Model > Add > Model Wizard : Perspective | Business Modeling | VDML
Other	⊙ Perspective ˙ > Business Modeling > VDML > Model Patterns