Enterprise Architect

User Guide Series

The Modeling Team

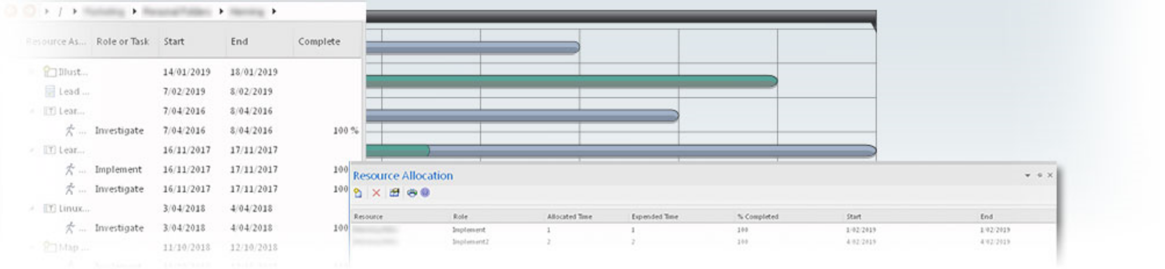
How to support modeling in teams? Sparx Systems Enterprise Architect is a team modeling platform, with a free readable Light version, many deployment options, version control, reusable assets, reviews, Team Library, user security and Workflow Scripts.

Author: Sparx Systems
Date: 2022-10-03
Version: 16.0

CREATED WITH ENTERPRISE ARCHITECT

Table of Contents

The Modeling Team



Resource As...	Role or Task	Start	End	Complete
Illust...		14/01/2019	18/01/2019	
Lead ...		7/02/2019	8/02/2019	
Lear...		7/04/2016	8/04/2016	
...	Investigate	7/04/2016	8/04/2016	100 %
Lear...		16/11/2017	17/11/2017	
...	Implement	16/11/2017	17/11/2017	100
...	Investigate	16/11/2017	17/11/2017	100
Linux...		3/04/2018	4/04/2018	
...	Investigate	3/04/2018	4/04/2018	100
Map ...		11/10/2018	12/10/2018	

Resource Allocation

Resource	Role	Allocated Time	Expended Time	% Completed	Start	End
[illegible]	Implement	1	1	100	1/02/2019	1/02/2019
[illegible]	Implement2	2	2	100	4/02/2019	4/02/2019

Enterprise Architect has been built from the ground up as a team modeling platform, and has extensive support for groups of people working together on the same projects, sharing information, ideas and models. Features in team support include Baselines, Version Control and a Reusable Asset Service, which protect the valuable modeling assets in a team environment, plus tools such as a Discussion Forum, Library window and Gantt Charts to facilitate collaboration between project members. The role-based security system has also been designed to encourage collaboration, allowing team members to work together confident that there will be no conflicts in accessing or changing model data.

A choice of deployment options will support any team development environment, allowing people to work centrally or remotely in highly distributed environments. Corporate policy and standards can also be built into the models with the use of Workflow Scripts. A free 'Lite' version of Enterprise Architect offers team members 'view only' access to their models, yet also allow them to generate high quality corporate documentation in a wide number of formats to communicate with people outside the modeling platform.

Overview

Facility	Description
Team Development	Set up a collaborative modeling environment, taking advantage of security, workflow and shared reference data, as discussed in the rest of this topic.
Formal Model Reviews	A simple yet effective mechanism for capturing, in real time, reviews of a section of the model in line with a particular event. Typically, a Project Manager or coordinator will create a Review element specifically to discuss one or more elements for a project phase or other category of review, over a defined period.
Project Management	Explore some of the ways you can manage your project and team within Enterprise Architect.
Project Resources	Track and manage the people and resources in your project.

Glossary	Define a common vocabulary between your different teams, ensuring common understanding.
Task Allocation	Assign and Track team tasks in a Gantt View.
Personal Tasks	Record and manage your personal tasks within the project.
Model Mail	Use Model Mail in the Chat & Mail window to securely communicate with your team via an internal email system embedded within the model.
Project Calendar	Track the deployment of resources, time-frames for tasks, and upcoming project events such as meetings and milestones, in a calendar format.
Use Case Estimation	Form an estimate of the complexity of a system and an indication of the effort required to implement the model.

Library Window	Provides access to a team-based library of documents to record and discuss the development and progress of the project.

Making Project Data Available in a Distributed Environment

Enterprise Architect offers a diverse set of functionality designed specifically for sharing projects in team-based and distributed development environments; for example: Cloud-based solutions, network deployment of model repositories, replication and Native/XMI Import/Export.

Applying Security to the Model

User Security is a means of improving collaborative design and development by preventing concurrent editing and inadvertent model changes by users not designated as model authors.

Using an Internal Discussion Forum

The Discussions facility provides several mechanisms to support your development team community, generally in discussing the development and progress of the model across the project, or specifically in discussing individual elements in the model, the discussions becoming a component of each element.

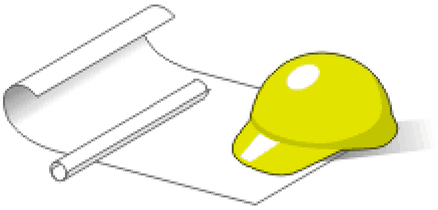
Building Company Policy and Project Development Guidelines into the Project

You can create workflow scripts that provide a robust approach to applying company policy and strengthening project development guidelines, by validating the work against the policy and procedures within the model itself.

Standardizing and Re-using Project Data

You can import and export Reference data (including Glossary and Issue information) from .XML files of another iteration of the same model, or of a different model.

Typical Project Roles



Enterprise Architect is an effective multi-disciplinary modeling platform that supports common work practices and provides features to assist the entire spectrum of roles and disciplines across enterprise, business, engineering and software projects. Each role will typically use different features of the tool. A number of the roles and their responsibilities that the system supports are outlined here.

You can review a summary of the typical tasks supported for each role, or review the Help topic for the appropriate role title to explore how Enterprise Architect can assist you in carrying out that role within a model-driven project.

Roles and Responsibilities

Role	Responsibilities
Business Analyst	Create high-level models of business processes.
Software Architect	Map functional requirements, perform real time modeling of objects, design the Deployment model and detail the

	deliverable components.
Software Engineer	Map Use Cases onto Class diagrams, detail the interactions between Classes, define the system deployment and define software Packages.
Developer	Perform round trip code engineering, including reverse engineering of existing code and generation of code from Class elements.
Project Manager	Assign resources to elements, measure risk and effort, estimate project sizes, and manage element status, change control and maintenance.
Tester	Create test scripts against elements in the modeling environment.
Implementation Manager	• Track and assign maintenance-related items to elements within Enterprise Architect • Rapidly capture and keep records of maintenance tasks such as features, changes, documents, issues, defects and tasks • Trace the maintenance of the items and processes involved in system

	deployment
Technology Developer	Create customized additions to the functionality already present within Enterprise Architect.
Database Developer	Develop databases, including modeling database structures, importing database structures from an existing database and generating DDL for rapidly creating databases from a model.

Summary of Typical Tasks

Throughout a design and development project there are many different tasks to be performed, which could be carried out either by one person or - more probably - by members of a team with different responsibilities. In either case, Enterprise Architect supports most - if not all - of the responsibilities you might have on your project. The descriptions in this topic identify a number of job roles that the system supports. For those that most resemble your role on a project, refer to the Help topic for that job title to read a description of how that role might make use of Enterprise Architect, then use the references within those topics to explore some of the features of importance to the role.

Summary of Typical Job Roles

Most of these roles work with specific types of diagram, so you might want to learn more about diagram types in general and specific types of diagram in particular.

Several types of project team member might want to generate documentation on their work and report on how the project is developing and changing. Using Enterprise Architect you can generate project reports in either document or web format.

Role	Responsibilities

Business Analyst	For modeling: • Requirements • High-level business processes • Business activities • Work flows • System behavior
Database Developer	• Developing databases • Modeling database structures • Creating logical data models • Generating schema • Reverse engineering databases
Software Architect	• Mapping functional requirements of the system • Mapping objects in real time • Mapping the deployment of objects • Defining deliverable components
Tester	• Developing test cases • Importing requirements, constraints and scenarios • Creating Quality Test documentation • Tracking element defects and changes
Software Engineer	• Mapping Use Cases into detailed Classes

	• Defining the interaction between Classes • Defining system deployment • Defining software Packages and the software architecture
Project Manager	• Providing project estimates • Resource Management • Risk Management • Maintenance Management
Developer	• Forward, reverse and round-trip engineering • Visualizing the system states • Visualizing Package arrangements • Mapping the flow of code
Implementation Manager	• Modeling the tasks in rolling-out a project, including network and hardware deployment • Assigning and tracking maintenance items on elements (issues, changes, defects and tasks)
Technology Developer	For creating or customizing: • UML Profiles • Patterns

	• Code Templates • Tagged Value Types • MDG Technologies • Add-Ins

Notes

- The Corporate, Unified and Ultimate Editions of Enterprise Architect have a User Security feature that can be applied or turned off; if security is turned on, you need to have the appropriate access permissions to use many of the facilities

Enterprise Architect

Enterprise Architects can use the tool to create a deep representation of an enterprise, including Business, Information, Application, and Technology architectures.

You can create Roadmaps as overlays on any diagram type, and capability models can ensure the architectures align with what the enterprise does. You can create Business, Information, Application, and Technology Architectures, and these and the teams that create them can be managed through the tool. Architects and other stakeholders can automatically generate business-friendly publications, including charts and graphs, to visualize critical parts of the architectures.

Enterprise Architect Tasks

Task	Detail
Create Strategic Roadmaps	Roadmaps guide an organization from its current state to a target state, thus transforming the enterprise and steering it on a strategic course through a series of transition states. Enterprise Architect has productive and flexible roadmap facility, allowing you to create roadmaps from pre-built model

	patterns for a solution, segment, and enterprise architectures. You can develop roadmaps for any architecture artifacts, including capabilities, applications, and technology items.
Define Business Capability Models	Business Capabilities are the cornerstone for the work carried out by the other architecture domains, including Information, Application, and Technology Architecture. They provide a way of viewing what the business does or needs to do. They are considerably easier and less time-consuming to create than business processes and directly correlate to what the enterprise does. You can model capabilities using ArchiMate capability elements or UML Activities and diagrams of nested elements modeling the hierarchal structure of these all-important business elements. The capabilities can be automatically colored in two dimensions using dynamic diagram legends.
Catalog Application Portfolios	The Application Architecture provides an essential catalog of the applications in the enterprise describing the work that they do to transform, transmit and store

	information. The architecture also describes the interfaces required or provided by the applications and how the applications interact to carry out the activities described in the business models, such as the Business Process diagrams. You can model application portfolios in Enterprise Architect and visualize the list in various ways, including diagrams, list views, specification manager, and charts and graphs.
Model Information Architectures	Information Architecture is key to the success of an Enterprise Architecture Program, as information is created, consumed, and destroyed by the components that make up the other architectures. The information architecture will typically include a description of the baseline and target architectures, with a series of transitions defined that can be executed and described on Roadmap diagrams. Enterprise Architect is a profoundly useful tool for creating and maintaining information architectures. Its sophisticated and extensive support for standards and its wide range of tools to

	support information models, from high-level classifications and concepts right down to the level of schemas and the elements and columns that comprise them.
Create Technical Reference Models	The Technical Reference Model (TRM) references generic platform services and technology elements and acts as a substrate upon which to build a technology architecture. The TRM provides a set of architectural and solution building blocks that will ultimately provide the platform for business and infrastructure applications to deliver the application and infrastructure services. You can use Enterprise Architect to create any technology model, and you can use any modeling language to represent the devices, nodes, system software, and any other technology artifact.
Manage Architecture Governance	The governance of the architectures is critical for the success of the program and the architectures it creates. Regardless of how perfect an architecture is, without the assurance that the technology staff

	has implemented it correctly, the vision expressed in the architecture might not be realized, the promise made to the stakeholders will be empty, and the business value will never eventuate. You can use Enterprise Architect to govern and manage your architectural practice and the architectures it produces, including governance boards, the governance register, and more.

Business Analysts

A Business Analyst can use Enterprise Architect to create high-level models of business processes, including business requirements, activities, workflow, and the display of system behavior.

Using Enterprise Architect, a Business Analyst can describe the procedures that govern what a particular business does. Such a model is intended to deliver a high-level overview of a proposed system.

Business Analyst Tasks

Task	Detail
Model High Level Business Processes	Using Analysis diagrams, you can model the high-level processes of the business. Analysis diagrams are a subset of UML 2.5 Activity diagrams and are less formal than other diagram types, but they provide a useful means for expressing essential business characteristics and requirements.
Model Requirements	Gathering requirements is typically the first step in developing a solution, be it for developing a software application or

	for detailing a business process; it is an important step in the implementation of a project. Using Enterprise Architect, you can define the Requirement elements, connect Requirements to the model elements for implementation, connect Requirements together into a hierarchy, report on Requirements, and move Requirements out of model element responsibilities.
Model Business Activities	You can use Activity diagrams to model the behavior of a system and the way in which these behaviors are related to the overall flow of the system. Activity diagrams do not model the exact internal behavior of the system but show instead the general processes and pathways at a high level.
Model Workflow	To visualize the cooperation between elements involved in the workflow, you can use an Interaction Overview diagram, which provides an overview of sub activities that are involved in a system.
Display System Behavior	In displaying the behavior of a system as a Use Case diagram, Enterprise Architect provides an easily understood tool for

	mapping the functional requirements and behavior of a system.

Systems Engineer

A Systems Engineer can use Enterprise Architect to produce robust and productive models of complex cyber-mechanical systems. The models can be requirements, structural and behavioral models, including Use Cases, Package Diagrams, Block Definition Diagrams, Internal Block Diagrams, Activity Diagrams, Sequence Diagrams, State Machine Diagrams, and Parametric Diagrams. Enterprise Architect has a comprehensive set of tools to assist the engineer and other stakeholders.

Systems Engineer Tasks

Task	Detail
Define and Manage Requirements and their Relationships	The field of Requirement Engineering is one of the most critical disciplines in the solution development lifecycle, and it has a documented impact on the success of projects. Enterprise Architect has an unparalleled range of tools for developing, managing, visualizing, and documenting requirements, including tools to import, integrate, and synchronize with external requirement management systems.

Describe User Goals with Use Cases	Systems Engineers use Use Cases as a method for representing functional requirements from the users' perspective. They are goal-driven because the Use Case defines the goal that the user is trying to achieve while interacting with the system. Enterprise Architect fully supports the development of Use Case diagrams and fully supports the modeling and management of Use Case text; it has a unique and highly productive tool for working with Use Cases, called the Scenario Builder.
Use Blocks to Model Structure and Constraints	The Block is the fundamental unit of system structure. Systems Engineers use blocks to describe an entire system, a subsystem, a component, an item that flows through a system, a constraint, or entities that reside outside a system. Similar to our natural languages, a Block can represent something abstract, logical, or physical. Enterprise Architect has a rich set of tools that help the systems engineer work with Blocks and visualize the structure and behavior of these all-important elements in a system's definition.

Coordinate Behavior with Activities	The Activity diagram is formally based on a branch of mathematics called Petri Nets, and it uses a system of tokens to indicate both the sequence of actions and also the items that flow through the system. The items that flow can be information items, physical items, or even control signals. Enterprise Architect provides a rich toolbox to work with these behavioral elements and their relationships, including allocating system behavior in the form of Activities and Actions to Blocks and relating these elements to behavioral features owned by Blocks, such as operations.
Visualize Systems in Motion with Simulations	Simulation provides a way to see a system in motion and visualize how it behaves through its lifecycle. As a leading Systems Engineering tool, Enterprise Architect allows a systems engineer to construct models using industry-compliant modeling techniques and languages to represent cyber-mechanical systems. These models act as devices for communication between collaborating engineers, consultants, and others but can also be

	used to generate visualizations and simulations using industry-standard modeling languages used by OpenModelica and MATLAB's Simulink.

Software Architects

Software Architects can use Enterprise Architect to map functional requirements with Use Cases, perform real time modeling of objects using Interaction diagrams (Sequence, Timing, Communication or Interaction Overview), design the Deployment model and detail the deliverable components using Component diagrams.

Software Architect Tasks

Task	Detail
Map Functional Requirements of the System	With Enterprise Architect you can take the high level business processes that have been modeled by the Business Analyst and create detailed Use Cases. Use Cases describe the proposed functionality of a system and are only used to detail a single unit of discrete work.
Map Objects in Real Time	You can use Interaction diagrams (Sequence and Communication diagrams) to model the dynamic design of the system. Sequence diagrams detail the messages

	that are passed between objects, and the lifetimes of the objects. Communication diagrams are similar to Sequence diagrams, but instead display the way in which the object interacts with other objects.
Map Deployment of Objects	You can use Deployment diagrams to provide a static view of the run-time configuration of processing nodes and the components that run on the nodes. Deployment diagrams show the connections between hardware, software and any middleware that is used on a system.
Detail Deliverable Components	Using Component diagrams, you can model the physical aspects of a system. Components can be executables, libraries, data files or another physical resource that is part of a system. The component model can be developed from scratch from the Class model or can be brought in from existing projects and from third-party vendors.

Software Engineers

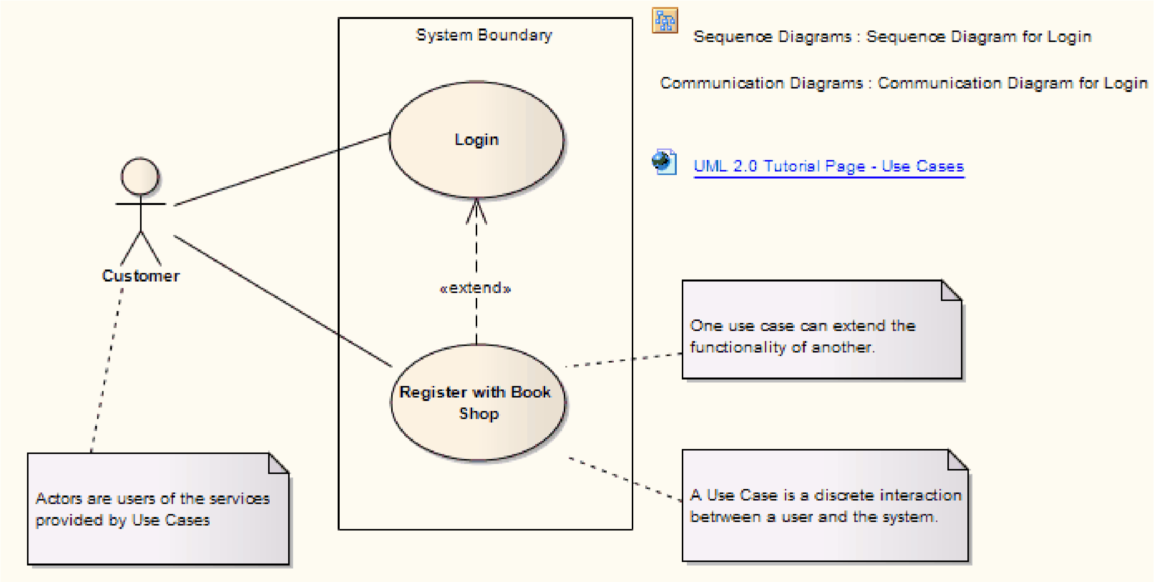
Software Engineers using Enterprise Architect can manually map Use Cases onto Class diagrams, detail the interactions between Classes, define the system deployment with Deployment diagrams and define software Packages with Package diagrams.

Software Engineering Tasks

Task	Detail
Map Use Cases into Detailed Classes	Within Enterprise Architect you can study the Use Cases developed by the Software Architect, and with that information create Classes that fulfill the objectives defined in the Use Cases. A Class is one of the standard UML constructs that is used to detail the pattern from which objects are produced at run time; to record the relationships between Use Cases and Classes, you can create diagrams linking the elements with Realization connectors, and/or map the Realization connectors in the Relationship Matrix.

Detail Interaction Between Classes	You can use Interaction diagrams (Sequence and Communication diagrams) to model the dynamic design of the system. Sequence diagrams are used to detail the messages passed between objects, and the lifetimes of the objects. Communication diagrams are similar to Sequence diagrams, but instead display the way in which objects interact with other objects.
Define System Deployment	Deployment diagrams provide a static view of the run-time configuration of processing nodes and the components that run on the nodes. Deployment diagrams can be used to show the connections between hardware, software and any middleware that is used on a system, to explain the connections and relationships of the components.
Define Software Packages	You can use Package diagrams to detail the software architecture. Package diagrams are used to organize diagrams and elements into manageable groups, declaring the dependencies.

Simple Use Case diagram


System Boundary
Login
«extend»
Register with Book Shop
Customer
Sequence Diagrams : Sequence Diagram for Login
Communication Diagrams : Communication Diagram for Login
UML 2.0 Tutorial Page - Use Cases
One use case can extend the functionality of another.
A Use Case is a discrete interaction between a user and the system.
Actors are users of the services provided by Use Cases

Developers

Developers can use Enterprise Architect to perform round trip code engineering, which includes reverse engineering of existing code and generation of code from Class elements.

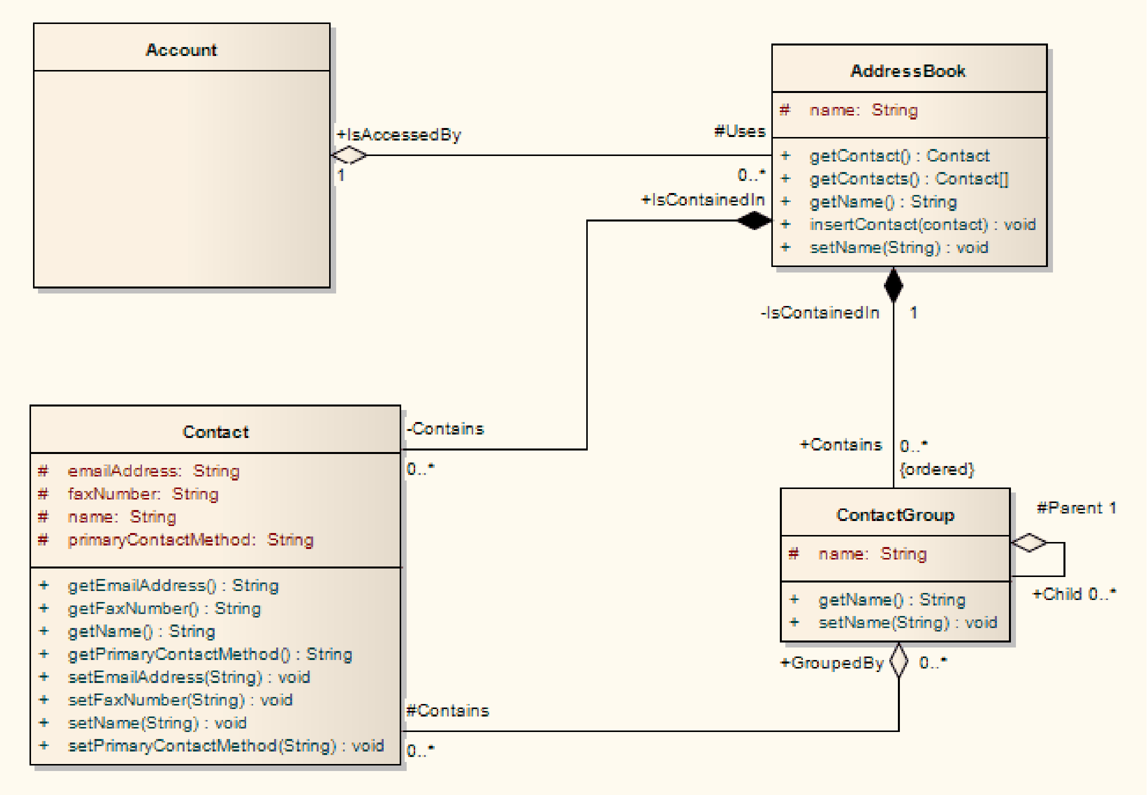
Developer Tasks

Task	Detail
Round Trip Engineering	Enterprise Architect gives you unparalleled flexibility in 'round tripping' software from existing source code to UML 2.5 models and back again. Round trip engineering involves both forward and reverse engineering of code, keeping the model and code synchronized.
Reverse Engineering	In Enterprise Architect, you can reverse engineer code from a number of supported languages and view the existing code as Class diagrams, which illustrate the static design view of the system. Class diagrams show Classes and interfaces, and the relationships between

	them; the Classes defined in UML Class diagrams can have direct counterparts in the implementation of a programming language.
Forward Engineering	As well reverse engineering your code into your model, you can forward engineer elements of your model into code (code generation). This way you can make changes to your model with Enterprise Architect and quickly implement the changes in the source code.
Determine the System State	To visualize the state of the system you can use StateMachine diagrams to describe how elements move between States, classifying their behavior according to transition triggers and constraining guards. StateMachine diagrams capture system changes over time, typically being associated with particular Classes; often a Class can have one or more StateMachine diagrams to fully describe its potential states.
Visualize Package	Package diagrams help you design the architecture of the system; they are used

Arrangement	to organize diagrams and elements into manageable groups, and to declare their dependencies.
Follow the Flow of Code	Activity diagrams help you develop a better understanding of the flow of code. Activity diagrams illustrate the dynamic nature of the system; you can model the flow of control between Activities and represent the changes in state of the system.

Simple Class Diagram



Notes

- You can use StateMachine, Package and Activity diagrams to better understand the interaction between code elements and the arrangement of the code

Project Managers

Enterprise Architect provides support for the management of projects. Project Managers can use the system to assign resources to elements, measure risk and effort, estimate project sizes, and manage element status, change control and maintenance.

Project Manager Tasks

Task	Detail
Provide Project Estimates	In Enterprise Architect you have access to a comprehensive project estimation tool that calculates effort from Use Case and Actor objects, coupled with project configurations defining the technical and environmental complexity of the work environment.
Resource Management	Managing the allocation of resources in the design and development of system components is an important and sometimes difficult task; Enterprise Architect provides you with an effective tool for assigning resources directly to model elements and tracking progress

	over time.
Risk Management	You can use the Risks window to assign risk to an element within a project; using risk types you can name the risk, define the type of risk and give it a weighting.
Maintenance	Within Enterprise Architect you can assign maintenance-related items to elements and track them, providing rapid capture and record keeping for items such as features, changes, documents, issues, defects and tasks. You can also create and maintain a Project Glossary of processes, procedures, terms and descriptions.

Testers

Enterprise Architect provides a design testing facility for Testers and Quality Assurance personnel to create a range of test scripts against elements in the modeling environment.

Testing Tasks

Task	Detail
Test Cases	With Enterprise Architect, you can set up a series of tests for each model element. The test types include Unit, Acceptance, System, Integration, Inspection and Scenario tests.
Import requirements, constraints and scenarios	To use testing to maintain the integrity of the entire business process, you can import requirements, constraints and scenarios defined in earlier iterations of the development life cycle. Requirements indicate contractual obligations that elements must perform within the model. Constraints are conditions that must be met in order to pass the testing process; constraints can be:

	- Pre-conditions (states that must be true before an event is processed) - Post-conditions (events that must occur after the event is processed) or - Invariant constraints (which must remain true through the duration of the event) Scenarios are textual descriptions of an object's action over time and can be used to describe the way a test works.
Create quality test documentation	Enterprise Architect provides the facility to generate high quality test documentation in .RTF, DOCX and PDF file formats.
Element defect changes	In defect tracking you can allocate defect reports to any element within the model, so that all who are involved in the project can quickly view the status of defects and see which defects have to be addressed and which have been dealt with.

Implementation Managers

Enterprise Architect provides support for the management of project implementation. You can track and assign maintenance -related items to elements within Enterprise Architect, and rapidly capture and update records of maintenance tasks such as features, changes, documents, issues, defects and tasks. By providing a centralized facility for each element involved in the deployment process Enterprise Architect offers a convenient solution for tracing the maintenance of the items and processes involved in system deployment.

Implementation Tasks and Tools

Task	Detail
Develop Deployment Diagrams	Using Deployment diagrams, you can model the roll out of a project, including network deployment and workstation deployment. Users involved in project deployment can add maintenance tasks to the diagram elements. Deployment diagrams provide a static view of the run-time configuration of nodes on the network or of workstations,

	and the components that run on the nodes or are used in the workstations.

Technology Developers

Technology Developers are Enterprise Architect users who create customized additions to the functionality already present within Enterprise Architect.

Additions include UML Profiles, Patterns, Code Templates, Tagged Value Types, Scripts, Custom Queries, Transformations, MDG Technologies and Enterprise Architect Add-Ins. By creating these extensions the Technology Developer can customize the Enterprise Architect modeling process to specific tasks and speed up development.

Developing Technologies

Extension	Detail
UML Profiles	By creating UML Profiles you can create a customized extension for building UML models that are specific to a particular domain. Profiles are stored as XML files and can be imported into any model as required.
Patterns	Patterns are sets of collaborating Objects and Classes that provide a generic template for repeatable solutions to

	modeling problems. As Patterns are discovered in any new project, you can publish the basic Pattern template. Patterns can be re-used with the appropriate variable names modified for any future project.
Code Templates	Code templates are used to customize the output of source code generated by Enterprise Architect; in this way you can generate code languages not specifically supported by Enterprise Architect and define how the system generates source code to comply with your own company style guidelines.
Tagged Value Types	Tagged Values are used in Enterprise Architect to extend the information relating to an element in addition to the information directly supported by the UML language. A Tagged Value, strictly, is the value of a property of a modeling item, the property being called a tag; for example: a Class element called Person might have a tag called 'Age' with the Tagged Value of '42'. More loosely, the combination of tag and

	value can be referred to as a Tagged Value. A Tagged Value Type is a group of parameters that define and/or limit the possible values of a tag and, in many instances, how a specific value is assigned to the tag; for example, the tag 'Age' might have a Tagged Value Type of 'Integer', so the user simply types in a numeric value. Alternatively, the type could be 'Spin', with lower and upper limits of, say, 20 and 120, so the user sets a value by clicking on arrows in the field to increment or decrement the value within the limits of 20 and 120. Typically, Tagged Values are used during the code generation process, or by other tools to pass on information that is used to operate on elements in particular ways.
MDG Technologies	MDG Technologies can be used to create a logical collection of resources that can contain UML Profiles, Patterns, Code Templates, Image files and Tagged Value types that are accessed through a technology file.
Enterprise	Using Add-Ins you can build your own

Architect Add-Ins	functionality into Enterprise Architect, creating your own mini programs that can extend the capabilities of the system, defining your own menus, and creating your own Custom Views.

Database Developers

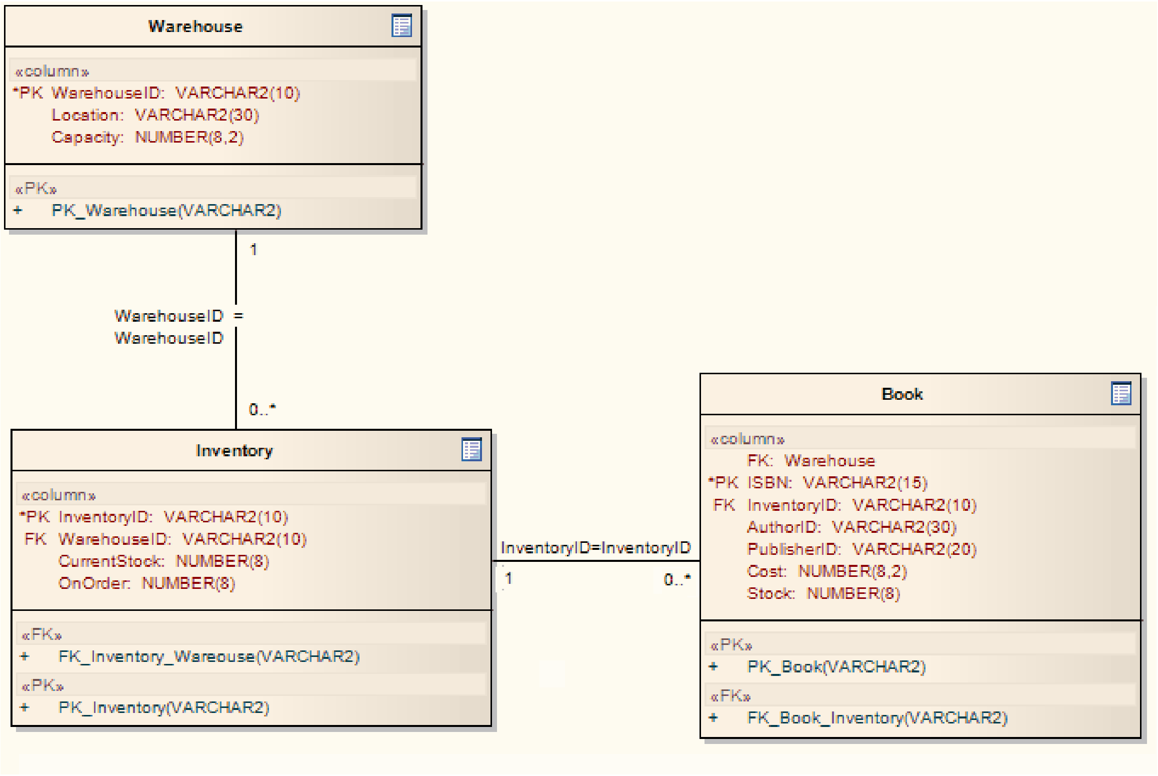
Enterprise Architect supports a range of features for the development of databases, including modeling database structures, importing database structures from an existing database and generating DDL for rapidly creating databases from a model.

Database Development Tasks

Task	Detail
Create Logical Data Models	With Enterprise Architect you can build database diagrams using the built-in UML Data Modeling Profile. This supports the definition of Primary and Foreign Keys, cardinality, validation, triggers, constraints and indexes.
Generate Schema	By using Enterprise Architect's DDL generation function you can create a DDL script to create the database table structure from the model. Enterprise Architect currently supports: • DB2 • Firebird

	• MS Access • MySQL • MS SQL Server • Oracle • PostgreSQL
Reverse Engineer Database	Using an ODBC data connection you can import a database structure from an existing database to create a model of the database. By generating the model directly from the database you can quickly document your work and create a diagrammatic account of a complex database through the graphical benefits of UML.

Example Data Model Diagram


Warehouse
«column»
*PK WarehouseID: VARCHAR2(10)
Location: VARCHAR2(30)
Capacity: NUMBER(8,2)
«PK»
+ PK_Warehouse(VARCHAR2)
1
WarehouseID =
WarehouseID
0..*
Inventory
«column»
*PK InventoryID: VARCHAR2(10)
FK WarehouseID: VARCHAR2(10)
CurrentStock: NUMBER(8)
OnOrder: NUMBER(8)
«FK»
+ FK_Inventory_Wareouse(VARCHAR2)
«PK»
+ PK_Inventory(VARCHAR2)
InventoryID=InventoryID
1
0..*
Book
«column»
FK: Warehouse
*PK ISBN: VARCHAR2(15)
FK InventoryID: VARCHAR2(10)
AuthorID: VARCHAR2(30)
PublisherID: VARCHAR2(20)
Cost: NUMBER(8,2)
Stock: NUMBER(8)
«PK»
+ PK_Book(VARCHAR2)
«FK»
+ FK_Book_Inventory(VARCHAR2)