

Sparx Systems

Enterprise Architect 7.0

Reviewer's Guide

Copyright Sparx Systems 2007



www.sparxsystems.com

Table of Contents

<i>Introduction</i>	4
<i>What is Enterprise Architect?</i>	4
<i>What differentiates Enterprise Architect from other UML tools?</i>	4
<i>How popular is Enterprise Architect now?</i>	4
<i>Summary of Enterprise Architect features</i>	5
<i>Some Enterprise Architect features in detail</i>	6
UML 2.1 Support	6
Model Validation	6
Code Engineering and Visualization	6
<i>Import .jar files and .NET assemblies</i>	7
<i>Template-driven source code generation</i>	7
<i>On-demand and live-generation of code</i>	7
<i>Built-in syntax highlighting source code with dynamic source 'outliner'</i>	8
Debugging, Compiling and Visualizing Executing Code	8
<i>Build and run scripts</i>	8
<i>Debugging Java, .NET and Microsoft Native executables using the Debug Workbench</i>	8
<i>Automatic generation of Sequence diagrams</i>	9
Built-in support for junit and NUnit testing	9
<i>Integrate unit testing into your modeling-development cycle</i>	9
Performance & Scalability	9
Version Control	9
Connectivity and integration with other tools	10
Automation Interface	10
Add-ins	10
<i>MDG Link</i>	10
<i>MDG Integration</i>	10
User interface, tools and productivity boosters	10
<i>Model patterns</i>	10
<i>User Interface</i>	10
<i>In-line editing of diagram elements</i>	11
<i>Quick linking technology</i>	11
<i>Other Diagramming features</i>	11
Traceability	11
<i>Auditing view</i>	11
<i>Element list</i>	12
<i>Track element use</i>	12
<i>Hierarchy (dependency) view</i>	12
<i>The Matrix view</i>	12
<i>Create diagrams with related elements</i>	12
Documentation and reporting in Enterprise Architect	12
<i>The Rich Text Format (RTF) report writer</i>	13
<i>The HTML report writer</i>	13
<i>Built-in document editor and linked documents</i>	13
Model Search	13
Other reports	13
Team development and collaboration	14
<i>Support for large models and many concurrent users</i>	14
XML import/export	14
Version control	14

Security	14
The model discussion forum	15
MDA (Model Driven Architecture) support	15
<i>Template driven transforms</i>	15
<i>Repeatable transforms</i>	15
<i>Built-in transforms</i>	15
<i>Adding new transforms</i>	16
Database modeling support	16
<i>Built-in data modeling profile</i>	16
<i>Support for major DBMS systems</i>	16
<i>DDL generation</i>	17
<i>Import of database structures from ODBC connections</i>	17
Project Management support	17
<i>Baselines and Differencing</i>	17
<i>Project estimation with Use Case metrics</i>	18
<i>Resources</i>	18
<i>Testing</i>	18
<i>Change management</i>	18
<i>Model Tasks</i>	18
<i>Glossary</i>	18
Import - Export Capability	18
<i>Variety of XMI formats</i>	19
<i>CSV</i>	19
Extending Enterprise Architect	19
<i>UML Profiles</i>	19
<i>UML Patterns</i>	19
<i>MDG Technologies</i>	19
<i>Custom Add-ins</i>	20
SOA (Service Oriented Architecture) support	20
<i>XSD Capability</i>	20
<i>WSDL Capability</i>	20
Business Process Modeling	20
<i>The BPMN Profile</i>	20
Requirements management	21
<i>Enterprise Architect Editions</i>	21
<i>Process support</i>	21
<i>About Sparx Systems</i>	22
Company vision	22
Ongoing commitment to Enterprise UML tools	22
Contact Details	22

Introduction

This document provides a comprehensive overview of the features and capabilities of Sparx Systems' **Enterprise Architect 7.0**. Each section focuses on a particular aspect of Enterprise Architect, providing an introduction to the purpose and benefits of each capability.

What is Enterprise Architect?



Enterprise Architect is a Computer Aided Software Engineering (CASE) tool for designing and constructing software systems, for business process modeling, and for more generalized modeling purposes.

Enterprise Architect is based on the latest UML 2.1 specification (*see* www.omg.org). UML defines a visual language that is used to model a particular domain or system (either proposed or existing).

Enterprise Architect is a progressive tool that **covers all aspects of the development cycle**, providing full traceability from the initial design phase through to deployment and maintenance. It also provides support for testing and change control.

What differentiates Enterprise Architect from other UML tools?

- Comprehensive UML 2.1-based modeling
- Built-in Requirements Management
- Extensive Project Management support, including resources, metrics and testing
- Testing support: test cases, junit and NUnit support
- Flexible documentation options: HTML and Rich-Text (RTF) report writers
- Code engineering support for many languages 'out of the box'
- An integrated Debug Workbench for profiling executable Java and .Net applications, instantiating run-time model objects and recording sequence diagrams from a stack trace
- Extendable modeling environment that can host user-defined profiles and technologies
- Usability
- Speed: Enterprise Architect is a spectacularly fast performer
- Scalability: Enterprise Architect can handle extremely large models and many concurrent users with ease
- Price: Enterprise Architect is priced to outfit the entire team, making collaboration and team development a real possibility.

How popular is Enterprise Architect now?

With over 100,000 licenses sold, Enterprise Architect has proven remarkably popular across a wide range of industries and is used by thousands of companies world-wide. From large, well-known, multi-national organizations to smaller independent companies and consultants, Enterprise Architect has become the UML modeling tool of choice for developers, consultants and analysts in over 60 countries.

Sparx software is used in the development of many kinds of software systems in a wide range of industries, including: aerospace, banking, web development, engineering, finance, medicine, military, research, academia, transport, retail, utilities (such as gas and electricity) and electrical

engineering. It is also used effectively for UML and business architecture training in many prominent colleges, training companies and universities around the world. Actual implementations range from single users to companies with over 600 seats working on large, distributed projects.

Summary of Enterprise Architect features

Enterprise Architect enables you to:

- Model complex software and hardware systems in UML-compliant notation
- Produce detailed and quality documentation in RTF and HTML formats
- Generate and reverse engineer ActionScript, C, C++, C#, Delphi, Java, PHP, Python, Visual Basic and VB.NET*
- Model databases and generate DDL scripts, and reverse database schema from ODBC connections*
- Manage change, maintenance and test scripts
- Model dependencies between elements
- Model system dynamics and state
- Model class hierarchies
- Model deployment, components and implementation details
- Collect project issues, tasks and system glossary
- Assign resources to model elements and track effort expended against required effort
- Export and Share models using the latest XMI 2.1 format. (Earlier versions also supported)
- Import models in XMI format from other tools
- Manage Version Control through XMI using SCC, CVS and Subversion configurations
- Use UML Profiles to create custom extensions for domain-specific modeling
- Save and load complete diagrams as UML patterns
- Analyze relationships between elements in tabular format using the Relationship Matrix
- Script and automate common tasks using a detailed Automation Interface
- Connect to database repositories on SQL Server 2000 and 2005, MySQL, Oracle9i and 10g, MS Access, PostgreSQL, Adaptive Server Anywhere, MSDE Server and Progress OpenEdge**
- Migrate changes across a distributed environment with JET Replication
- Use Controlled Packages based on XMI import and export
- Perform MDA Style Transforms.*

* Professional and Corporate Editions Only

** Corporate Editions Only

Some Enterprise Architect features in detail

Each of the following sections focuses on a particular aspect of Enterprise Architect's capabilities and defines the purpose and benefits of each capability.

UML 2.1 Support

Enterprise Architect supports all UML 2.1 models and diagrams. You can model business processes, web sites, user interfaces, networks, hardware configurations, messages and many other aspects of your development.

In brief, Enterprise Architect:

- Was the first UML tool to introduce comprehensive UML 2 support in April 2004
- Continues to refine and update UML 2 support
- Supports all 13 diagram types from UML 2

Readers unfamiliar with UML can find a short tutorial on the following URL:

http://www.sparxsystems.com/UML_Tutorial.htm.

Enterprise Architect provides additional diagram types that extend the core UML diagrams for business process modeling, mind mapping, formal requirements specifications and other domain-specific models. The modeling environment also provides a number of alternative views that make editing the core UML diagrams more intuitive and effective. One example is the State Table editor, which renders a standard UML State Machine diagram as an editable logic-table.

Model Validation

Model Validation checks UML models against known UML rules, as well as any constraints defined within the model, using the Object Constraint Language (OCL). You can perform Model Validation against a single UML element, a diagram or an entire package, as described below.

Validating a UML:

- **Element** validates the element and its children, features (attributes and operations) and relationships (connectors and links)
- **Diagram** validates the diagram itself (for correctness) as well as any elements and connectors within it
- **Package** validates the package, sub-packages and all elements, connectors and diagrams within it.

Code Engineering and Visualization

The Code Engineering process encompasses automated code generation, reverse engineering of source code and synchronization between the source code and model. It is available only in the Professional and Corporate editions of Enterprise Architect.

Enterprise Architect enables you to **generate source code from UML models** in more than ten commonly used development languages out of the box, including:

- ActionScript (Macromedia Flash development language)
- C
- C# (for both .NET 1.1 and .NET 2.0)
- C++ (standard plus .NET managed C++ extensions)

- Delphi
- Java (including Java 1.5, Aspects and Generics)
- PHP
- Python
- Visual Basic
- Visual Basic .NET

The source code generated includes class definitions, variables and function stubs for each attribute and operation in the UML class. In addition, Enterprise Architect enables you to edit and generate XSD and WSDL XML source (see the *SOA* section of this document).

Generating code is also known as **Forward Engineering**. Importing source code into model elements is known as **Reverse Engineering**. The languages that are available for code generation with Enterprise Architect are also available for reverse engineering.

Synchronization is when changes in the model are exported to the source and changes to source are imported into the model. This enables you to keep your model and your source code up-to-date as the project develops.

Round trip engineering is a combination of reverse and forward generation of code and should include synchronization in all but the most trivial of code engineering projects.

Import .jar files and .NET assemblies

Enterprise Architect (Professional and Corporate editions) enables you to reverse-engineer the following types of binary modules:

- Java Archive (.jar)
- .Net PE file (.exe, .dll); native Windows DLL and Exe files are not supported, only PE files containing .Net assembly data
- Intermediate Language file (.il).

Template-driven source code generation

You use Enterprise Architect's **Code Template Framework** (CTF) during the forward engineering of UML models. Code templates specify a customizable transformation from UML elements to the various parts of a given programming language.

The **Code Template Framework** enables you to:

- Generate source code from UML models
- Customize the way in which Enterprise Architect generates source code
- Forward engineer languages not specifically supported by Enterprise Architect.

The Code Template Framework consists of:

- Default Code Templates that are built into Enterprise Architect for forward engineering supported languages
- A Code Template Editor for creating and maintaining User-defined Code Templates.

On-demand and live-generation of code

Enterprise Architect provides a Live Code Generation feature that automatically updates your source code instantly as you make changes to your model. For example, when you create new

operations and attributes for a class in the model, these are instantly written out to the source file.

Built-in syntax highlighting source code with dynamic source ‘outliner’

You can use the built-in source code editor to view and modify any source code to open. If you select an element in a model and it has an associated source file, its code is shown in an editor with appropriate syntax highlighting and a navigable structure outline. The source code viewer also provides a toolbar for quickly generating code and synchronizing with the model.

Debugging, Compiling and Visualizing Executing Code

The Professional and Corporate versions of Enterprise Architect enable you to build, test, debug, run and execute deployment scripts, all from within the Enterprise Architect development environment.

This capability provides developers with tools to integrate UML development and modeling with source development and compilation. With the ability to generate NUnit and JUnit test classes from source classes using MDA Transformations and integrate the test process directly into the Enterprise Architect IDE, you can now integrate UML and modeling into the build/test/execute/deploy process.

In addition to build/test and execute functionality, Enterprise Architect includes debugging capabilities for Java, .NET and Microsoft Native (C++, C and VB). The debuggers built into Enterprise Architect are specifically designed to enable a developer or tester to capture stack trace information as they ‘walk through’ the executing code, performing runtime inspection of suspended threads. They can then use the final stack trace history to generate Sequence diagrams within Enterprise Architect, converting the actual code execution and calls into visual diagrams. This capability provides an excellent means of managing complexity within a project, of documenting existing code and ensuring that the code written performs as intended by the original architect or developer.

Build and run scripts

You can set up instructions for how to build and run the code behind any package. Each package can contain multiple configurations so that you can handle multiple versions with ease. Enterprise Architect can even parse the results of the compiler of your choice and open the source editor to any compile errors found.

Debugging Java, .NET and Microsoft Native executables using the Debug Workbench

Enterprise Architect enables you to debug source code within Enterprise Architect, using a debug interface appropriate to the source language. You configure debugging by creating a debug script for the packages to be tested. One of the primary objectives of this feature is to enable you to perform a debug ‘walkthrough’, executing code and capturing your stack trace for direct conversion into a Sequence diagram. This is a great way to document and understand what your program is doing during its execution phase.

System Requirements:

- Enterprise Architect 6.1 and later, supporting debugging .Net 1.n, .Net 2.0
- Java 2 Platform Standard Edition (J2SE) version 5.0
- Java 2 Enterprise Edition (J2EE) JDK 1.4 and above
- Native code compilation with a Microsoft compiler where an associated PDB file is available.

Automatic generation of Sequence diagrams

You can easily create detailed and comprehensive Sequence diagrams from your recorded debug sessions. Enterprise Architect makes it simple to either manually step through your executing code and record specific execution traces, or let the debugger automatically step through.

Enterprise Architect can take the recorded stack history captured during one of these runs and automatically build the Sequence diagram, including compacting looping sessions for easy reading.

Built-in support for jUnit and NUnit testing

Enterprise Architect provides two-fold support for unit testing with jUnit and NUnit. It helps you to create test classes and methods for your code with the jUnit and NUnit transformations. When the results of these transformations are generated you have test stubs that you only have to fill in with the testing logic.

The resulting code can then be compiled and executed using the Build and Run commands. Alternatively, you can provide a test script with the original package that runs your unit testing program. Enterprise Architect can capture the results of a command line execution and enable you to see where anything fails, all without leaving Enterprise Architect.

Integrate unit testing into your modeling-development cycle

One of the key principles of unit testing is that you should always write your tests first. Enterprise Architect helps you to do this. When you add a new method to a class, run the transformation on the class. Enterprise Architect creates a corresponding test method that you can generate and fill in the details. This can easily happen before the code you are testing has been created.

Performance & Scalability

Fast! Enterprise Architect users agree—Enterprise Architect loads in a fraction of the time of many tools currently on the market and scales readily to extremely large models without noticeable performance reductions.

Version Control

Enterprise Architect supports version control of packages and their component sub-packages to a central repository. This repository is maintained by third-party version control applications that control access and record revisions.

The Version Control feature of Enterprise Architect provides two key benefits:

- It provides the ability to coordinate sharing of packages between users
- It saves a history of changes to Enterprise Architect packages, and enables you to retrieve previous versions.

Enterprise Architect supports the following version control applications:

- Any version control product that complies with the Microsoft Common Source Code Control standard, version 1.1 or higher. (For example Visual Source Safe or Clear Case)
- Microsoft Team Foundation Server (TFS)
- Subversion, which is available from <http://subversion.tigris.org>
- CVS, which is available from <http://www.wincvs.org>

Connectivity and integration with other tools

Enterprise Architect provides a number of mechanisms for integrating your model with third-party tools, including a programmable API, an add-in framework and off-the-shelf Model Driven Generation solutions provided by Sparx.

Automation Interface

The Automation Interface enables you to access the internals of Enterprise Architect models. Here are some examples of tasks you could perform using the Automation Interface:

- Perform repetitive tasks, such as update the version number for all elements in a model
- Generate code from a state machine diagram
- Produce custom reports
- Ad hoc queries.

All development environments capable of generating ActiveX Com clients should be able to connect to the Enterprise Architect Automation Interface.

Add-ins

Add-ins enable you to add functionality to Enterprise Architect. The Enterprise Architect Add-in framework builds on the features provided by the Automation Interface to enable you to extend the Enterprise Architect user interface. Add-ins provide several key advantages over stand-alone automation clients, including the ability to define additional Enterprise Architect menus and receive notifications about various Enterprise Architect user-interface events, such as menu-clicks and user selections.

MDG Link

Sparx has developed a number of MDG products to provide interoperability with other tools. MDG Link products exemplify use of the Add-in framework to extend the functionality of Enterprise Architect. MDG Link for Visual Studio and MDG Link for Eclipse enable Enterprise Architect to interoperate with the Microsoft® Visual Studio® 2003 and Eclipse IDEs respectively.

MDG Integration

MDG Integration for Visual Studio 2005 tightly integrates Enterprise Architect into the Microsoft® Visual Studio® 2005 development environment. This product enables users to explore and edit the UML model inside Visual Studio and also provides many of the key features of Enterprise Architect directly within the IDE, including rich text and web-based document generation, MDA transformations, Baseline management and Engineering of key XML-based technologies.

User interface, tools and productivity boosters

Model patterns

Enterprise Architect ships with a number of predefined Model Patterns to assist in the creation of new Projects and Models. Each pattern contains useful notes, references and starter elements that together provide a framework on which you can build your model.

User Interface

The Enterprise Architect User Interface consists of a range of high-impact windows, menus and

toolbars that can be hidden if not required or displayed in whatever position on the screen is convenient for your work methods. Key amongst these are:

- The Project Browser, which displays the complete contents of your model or project in a hierarchical format (with optional level numbering), and which enables you to add, select, reorganize or delete packages, diagrams and elements anywhere in the project.
- The Enterprise Architect UML Toolbar, which is context-sensitive to the diagram being created, and provides a quick and efficient means of selecting and creating the appropriate model elements or connectors, whether they be UML, from extended diagrams or from imported technologies
- The Diagram View, which enables you to display and develop diagrams selected from the Project Browser; diagram backgrounds, connectors and elements can be coloured with or without a color gradient for color-coding or for better display and presentation.
- Context menus, which provide options specific to the type of object and its immediate environment.

In-line editing of diagram elements

Enterprise Architect enables fast editing of element properties directly from the diagram. Using in-place edit commands, you can rapidly add and update element features such as attributes, operations and parameters, all without leaving the Diagram View.

Quick linking technology

The Quick Linker provides a fast and 'in-place' mechanism for creating new elements and connectors on a diagram. Its context-sensitive selection menus help guide creation of 'correct' models, saving users' time and improving overall productivity.

Other Diagramming features

- You can export diagrams to a range of image formats (.bmp, .jpg, .png, .gif, .emf and .wmf)
- Swimlanes enable logical partitioning of diagrams
- The Pan & Zoom window enables easy navigation and preview of complex diagrams
- You can lock down diagrams to prevent accidental modification
- Enterprise Architect Shape Scripts enable you to specify custom shapes via a scripting language, each shape being keyed to a stereotype; these custom shapes are drawn instead of the standard UML notation for every element of the same stereotype
- You can also superimpose alternative images – as metafiles - on diagrams and elements, to replace the standard image.

Traceability

Auditing view

Enterprise Architect's Auditing feature enables you to track and record changes made to the model over time. By enabling this option, model administrators can view a range of information regarding changes, such as:

- Who changed an element
- How many elements they changed

- When they changed the data
- What the previous values were, and
- What type of elements they changed.

The Audit View can be tailored to show changes of specific types (including the changes to settings of the Audit View itself), to specific areas or levels of the model, over specific time periods and by each user. The Audit View can be synchronized with the Project Browser and Element List (see below) to check for changes as you review the elements, and those changes can be automatically displayed in an Audit History in the Enterprise Architect Output window.

Element list

The Element List is a tabular, editable view of elements that can be displayed in the main workspace. You can use the Element List to streamline the process of creating and updating elements in a package or diagram selected from the Project Browser window. This can be particularly useful for analysts to create and maintain formal requirement definitions within the model. You can also print the list or generate an RTF document directly from the entries on the Element List.

Track element use

Enterprise Architect makes it easy to track and display the use of an element. The ‘Show Usage’ feature in a diagram, the Project Browser or the Element List shows all occurrences of a given element throughout the model, and enables you to easily navigate to any occurrence.

Hierarchy (dependency) view

The Hierarchy window shows a mini picture of the current element's composition with respect to other elements. This information is derived from relationships with child or related classes. Relationships shown in the hierarchy include aggregation, inheritance and dependency; embedded elements are also shown. This helps extend the picture of where an element exists in the model space.

The Matrix view

The Relationship Matrix enables you to study the relationships between model elements in a tabular view. It also enables you to create, modify and delete relationships between elements with a single mouse click.

Create diagrams with related elements

Enterprise Architect can automatically populate a diagram with all elements that relate to a given element. You can filter the inserted elements based on the type, direction and depth of the relationship. The ‘Insert Related Element’ feature provides a fast and powerful way to build up specific relationship overviews for your frameworks or reverse engineered source code.

Documentation and reporting in Enterprise Architect

The production of documentation is essential to realizing the full benefit of Enterprise Architect. Enterprise Architect outputs high quality documentation in either RTF or HTML format. You can modify the RTF formatting directly with RTF Style templates to alter the look and feel of generated output. Using MS Word you can further enhance the output by connecting and interweaving model output in linked documents.

There are many ways to specify the Enterprise Architect content being documented. You can:

- Document a package and/or its child packages by manually highlighting the package and selecting a documentation control
- Specify embedded packages for exclusion if child packages are recursively documented
- Link a package to an RTF document template to simplify generating consistent types of documentation (such as Use Case reports) using the Documents feature.

You can select, group and order packages together in a manner different from the project view by creating 'virtual' documents.

The Rich Text Format (RTF) report writer

The RTF Style Template editor enables you to create and edit custom RTF templates to define any output RTF documentation. The Style Template Editor enables you to select particular model elements and then to specify, from the element type, the fields for inclusion in the generated document.

Formatting styles can be defined in the Style Editor, and items such as tables of contents and headers can be added to the document.

The HTML report writer

Enterprise Architect enables you to export an entire model or a single branch of the model to HTML web pages. The HTML report provides an easy-to-use and highly detailed model tree. In addition, hyperlinked elements make browsing to related information very easy. The HTML documentation is based on user-customizable HTML templates, so you can tailor the generated web-pages to suit your company standard.

Built-in document editor and linked documents

Enterprise Architect enables you to link rich-text documents to any element in the model. Linked documents are created from customizable templates and are included in generated web and Word-based reports.

Model Search

The Model Search generates a report list that you can view in the main workspace. It lists each element in the model that meets the highly versatile criteria you define within the search terms and search type. The elements listed in the search results are selectable for printing, reporting, editing, adding to documentation, and inserting into discussion forum topics.

Other reports

Enterprise Architect supports a variety of useful reports out-of-the-box including:

- Resource and Tasking Details
- Project Issues
- Project Glossary
- Project (size) statistics
- Dependency and Implementation Details
- Testing Details.

Team development and collaboration

Enterprise Architect offers a diverse set of functionality designed specifically for sharing projects in team-based and distributed development environments. Projects can be shared through network deployment of model repositories, replication, XMI Import/Export, Version Control, Package Control and User Security.

Network deployment can be achieved with an EAP (file based) repository or by using a dedicated DBMS (server based) repository.

Support for large models and many concurrent users

The Corporate Edition of Enterprise Architect enables the use of database repositories rather than use the standard .EAP files to store model data. Enterprise Architect supports the following DBMS repositories:

- MS SQL Server
- MySQL
- Oracle
- PostgreSQL
- Progress OpenEdge
- MSDE Server
- Adaptive Server Anywhere.

XML import/export

You use the XMI Import/Export facility to model discrete packages that can be exported and shared between developers. XMI enables you to export packages into XML files that can then be imported into any model.

Package control can be used to set up packages for version control and enables batch export of packages using XMI. Version Control enables you to maintain a repository by a third-party source code control application, which is used to control access and record revisions.

Version control

Enterprise Architect supports version control of project root nodes, packages and their component sub-packages to a central repository. This repository is maintained by third-party version control applications that control access and record revisions. Version control products supported for Enterprise Architect include CVS, Subversion, Microsoft Team Foundation Server (TFS) and SCC-compliant tools (including Visual Source Safe and Clear Case).

Security

User security in Enterprise Architect can be used to limit access to update functions within the model. Elements can be locked per-user or per-group; where user security is enabled a password is required to log in to the model. Security in Enterprise Architect is not designed to prevent unauthorized access; rather it is intended as a means of improving collaborative design and development by preventing concurrent editing and limiting the possibility of inadvertent model changes by users not designated as model authors.

User security is available only in the Enterprise Architect Corporate edition. User security in Enterprise Architect offers two policies: the standard security model and the rigorous security

model. In the standard security model all elements are considered unlocked and, as the need arises, a user can lock any element or set of elements at the user or group levels depending on permission rights that the user has to the model. The rigorous security model assumes that everything in the model is locked until explicitly checked out with a user lock. In this mode, an Enterprise Architect model is read-only until a user applies an editing lock on one or more elements. For more detailed information regarding the security policies view the security policy topic.

The model discussion forum

The Project Discussion Forum enables users to discuss the development and progress of a project. Team members can view and post messages within the modeling environment and can link their posts directly to elements within the model. For distributed team environments, users can connect their Enterprise Architect model to a Project Discussion Forum residing on a remote server.

MDA (Model Driven Architecture) support

Enterprise Architect provides the capability of executing MDA transforms. It provides a fully configurable method of converting model elements and model fragments from one domain to another. This typically involves converting Platform-Independent Model (PIM) elements to Platform-Specific Model (PSM) elements. A single element from the PIM could be responsible for creating multiple PSM elements across multiple domains.

Transformations are a huge productivity boost, and reduce the need to manually implement stock classes and elements for a particular implementation domain; for example, database tables can be automatically derived from persistent PIM classes. Enterprise Architect includes some basic built-in transformations, such as PIM to Data Model, PIM to C#, PIM to Java and PIM to XSD.

Template driven transforms

You can define a transformation using Enterprise Architect's simple code generation template. The process involves no more than writing a template to create a simple intermediary source file. Enterprise Architect reads the source file and binds that to the new PSM. All the templates are editable, so you can customize any given transformation to suit your needs.

Repeatable transforms

The transformations provided by Enterprise Architect provide the repeatability that is necessary for reliable software development. You can ensure that all components of your system are kept consistent. Enterprise Architect even enables you to handle more complex procedures through the facility to automatically generate code and perform further transformations on the result of any transformation.

Built-in transforms

Enterprise Architect comes with some built-in transformation types. These transformations are designed to cater for the requirements of as many users as possible, to be a good base to modify to include the specifics of your custom domain, and to be good examples of how to write transformations.

The following transformations are included in Enterprise Architect:

- C#
- DDL
- EJB Entity

- EJB Session
- Java
- JUnit
- NUnit
- WSDL
- XSD.

Adding new transforms

Enterprise Architect makes it easy to define and share transformations in addition to the ones provided by Sparx Systems. This means that you are not limited to the transformations that are already provided.

Database modeling support

Built-in data modeling profile

Enterprise Architect's built-in Data Modeling profile extends to the UML to provide an intuitive mapping from the database concepts of tables and relationships onto the UML concepts of classes and associations. These extensions also enable you to model database keys, triggers, constraints, RI and other relational database features.

When modeling or designing databases you might typically:

- Create a Data Model diagram
- Create a table
- Set properties of a table
- Create columns
- Create primary keys
- Create foreign keys
- Create stored procedures
- Create indexes, sequences, functions and triggers
- Generate DDL for a table
- Generate DDL for a package
- Convert datatypes for a table
- Convert datatypes for a package
- Customize datatypes for a DBMS
- Import a database schema from an ODBC data source
- Create views

Support for major DBMS systems

Enterprise Architect supports data modeling of database schema from the following databases:

- DB2

- InterBase
- Informix
- Ingres
- MS Access
- MySQL
- Oracle 9i and 10g
- PostgreSQL
- MS SQL Server 2000 and 2005
- SQL Server7
- Sybase Adaptive Server Anywhere
- Sybase Adaptive Server Enterprise
- Firebird.

DDL generation

Enterprise Architect can automatically generate DDL scripts based on your data model. Enterprise Architect's DDL generation capability supports 9 DBMS targets out of the box, with a range of options for customizing the generated output.

Import of database structures from ODBC connections

Analysis of legacy database systems is possible using Enterprise Architect's reverse engineering capabilities. By connecting to a live database via ODBC, you can import the database schema into a standard UML model. Subsequent imports allow you maintain synchronization between the data model and the live database.

Project Management support

Enterprise Architect provides support for managing your project. Project managers can use Enterprise Architect to assign resources to elements, to measure risk and effort and to estimate project size. Enterprise Architect also supports change control and maintenance.

Baselines and Differencing

The Enterprise Architect Corporate edition provides a facility to 'Baseline' (snapshot) a model branch at a particular point in time. Baselines are in XMI format and are stored within the model in compressed format. More than one baseline can be stored against a single Enterprise Architect package. Using Baselines, you can take a snapshot of a model branch at a particular point in development for later comparison to the current package state.

The Compare (diff) Utility in the Professional and Corporate editions enables you to explore what has changed within a model over time and how previous versions of a model branch differ from what is currently in the model. This utility enables you to compare a model branch in Enterprise Architect with:

- A Baseline created using the Baseline functionality
- A file on disk, created previously using the Enterprise Architect XMI export facility, or
- The current version-controlled XMI file on disk as created when using Version Control in Enterprise Architect.

Project estimation with Use Case metrics

The Use Case Metrics capability within Enterprise Architect makes it easy to assess the relative complexity of a software project, based on the number and type of use cases within the model, and on the type of development project and the capabilities of the development environment. With experience, the use case metrics approach is a great way to quickly assess the scope of a project.

Resources

Enterprise Architect can store resource and basic development information along with the model. This approach helps to create connection between the Project Manager and the development team, allowing for up-to-the-minute reporting and assessment of how a project is developing. What 'hot spots' have occurred, what elements are overdue and other critical resource issues can be tied directly to model elements and searched with ease using the comprehensive search facility within Enterprise Architect.

Testing

In addition to the integrated jUnit and NUnit testing capabilities, Enterprise Architect also enables you to attach arbitrarily complex tests to any model element. Keeping the model elements and the testing documentation in one integrated model significantly improves the communication between the test-team and the software developers and architects. Again, the detailed search facilities make it easy to find failing test cases, test cases not run and tests cases that have been passed. Using the testing and search capabilities, it is easy to navigate through the model and quickly locate problem spots, design flaws and other critical issues. Enterprise Architect is not only a UML Modeling environment; it is also a complete Test Management environment.

Change management

As projects develop and mature, one of the critical issues is change management. Keeping track of incremental changes is essential to managing the overall development process. Again, Enterprise Architect supports change management by enabling you to track meta-information about changes, issues, features and requirements against actual UML model elements.

Model Tasks

Tracking, prioritizing and assigning tasks is a critical part of managing a model's development over time. In addition to the other project management capabilities, Enterprise Architect also supports tracking model tasks against the project as a whole.

Glossary

As an additional feature Enterprise Architect adds a model glossary to each development project, making it simple to define and disseminate new terms and definitions, possibly unfamiliar to team members new to the project or the problem domain.

Import - Export Capability

Exchanging data between different models, and between different tools, is an integral part of any development project. Enterprise Architect supports both of the two most common data exchange formats (XMI and CSV), making it easy for information and models to be ported into different tools and repositories as needed. Coupled with the Enterprise Architect API, this makes it possible to automate model import/export, and facilitates the easy dissemination of models and automated software construction.

Variety of XMI formats

Enterprise Architect supports import and export in a wide range of XML Model Interchange (XMI) based formats. XMI is a specification for how to render complex model information to human-readable XML, for the general purpose of exchanging information with other tools. XMI is an open standard managed by the OMG. Enterprise Architect supports XMI 1.0, XMI 1.1 and XMI 2.1. Support for a range of formats is essential, as many tools expect a specific XMI version.

CSV

In addition to XMI import/export, Enterprise Architect offers a simple CSV data import/export facility. This is useful for outputting information to tools such as Microsoft Excel, and for importing similar lists of elements stored in spreadsheets.

Extending Enterprise Architect

UML Profiles

UML Profiles provide a means for extending the UML Language, which enables you to build UML models in particular domains. The profiles are based on additional stereotypes and tagged values that are applied to Elements, Attributes, Methods, Links and Link Ends. A profile is a collection of such extensions that together describe some particular modeling problem and facilitate modeling constructs in that domain. For example, the UML Profile for XML defined by David Carlson in *Modeling XML Applications with XML* (p. 310) describes a set of extensions to basic UML model elements to enable accurate modeling of XSD Schemas.

Enterprise Architect has a generic UML Profile mechanism for loading and working with different Profiles. UML Profiles for Enterprise Architect can be imported into Enterprise Architect in the Resource page of the Project Browser. Once you have imported the Profile elements, you can drag and drop them onto the current diagram. Enterprise Architect attaches the stereotype, tagged values and default values, notes and even the metafile if one is specified, to the new element. You can also drag and drop attributes and operations onto existing classes and have them immediately added with the specified stereotype, values and other parameters.

UML Patterns

Patterns are parameterized collaborations; that is, they are a group of collaborating objects and classes that can be abstracted from a general set of modeling scenarios. Patterns are an excellent means of achieving re-use and building in robustness. As patterns are discovered in any new project, the basic pattern template from previous engagements can be re-used with the appropriate variable names modified for the current project.

Patterns generally describe how to solve an abstract problem, and it is the task of the pattern user to modify the pattern elements to meet the demands of the current engagement.

MDG Technologies

The Model Driven Generator (MDG) Technologies enable you to download a logical collection of resources pertaining to a specific technology from a directory folder or web site into one centralized location in Enterprise Architect. With MDG Technologies you also have the option of granular importation of UML Profiles, UML Patterns, Code templates and Language types as an XML file into a single, easy to access area contained in the Enterprise Architect Resources window.

To get you started with MDG Technologies, Sparx Systems provide technologies such as ICONIX and Mind Mapping with the Enterprise Architect Installer, and offer further technologies for download from the Sparx systems website (http://www.sparxsystems.com/resources/mdg_tech).

Custom Add-ins

Add-ins enable you to extend Enterprise Architect's user interface and provide several advantages over stand-alone automation clients:

- Add-ins can define Enterprise Architect menus and sub-menus
- Add-ins receive notifications about various Enterprise Architect user-interface events including menu clicks and file changes
- Add-ins can (and should) be written as in-process (DLL) components; this provides lower call overhead and better integration into the Enterprise Architect environment
- Because a current version of Enterprise Architect is already running there is no need to start a second copy of Enterprise Architect via the automation interface
- Because the add-in receives object handles associated with the currently running copy of Enterprise Architect, more information is available about the current user's activity, such as which diagram objects are selected
- You are not required to do anything other than install the add-in to make it usable; that is, you do not have to configure add-ins to run on your systems.

SOA (Service Oriented Architecture) support

Enterprise Architect enables you to rapidly model, forward engineer and reverse engineer two key W3C XML technologies: XML Schema (XSD) and Web Service Definition Language (WSDL).

XSD and WSDL support is critical for the development of a complete Service Oriented Architecture (SOA), and the coupling of UML 2.1 and XML provides the natural mechanism for specifying, constructing and deploying XML-based SOA artifacts within an organization.

XSD Capability

XML schemas are modeled using UML class diagrams and the XML Schema toolbox. Enterprise Architect's XML Schema toolbox provides in-built support for the UML profile for XSD. This enables an abstract UML class model to be automatically generated as a W3C XML Schema (XSD) file.

WSDL Capability

Enterprise Architect supports forward and reverse engineering of the W3C Web Service Definition Language (WSDL). Enterprise Architect's WSDL toolbox can be used to conveniently model WSDL documents, which are represented as components marked with the stereotype *WSDL*. WSDL documents are contained in a package hierarchy representing the target WSDL namespace and its constituent XSD types, messages, port types, bindings and services.

Business Process Modeling

There are a number of approaches to Business Process Modeling (BPM) using UML as the base modeling language. In particular, Activity diagrams, Object Diagrams and custom Profiles provide a wealth of modeling possibilities for BPM analysts to use. Enterprise Architect compliments the basic UML 2.1 language palette with custom analysis, requirements management and process management elements (such as change, feature and issue elements).

The BPMN Profile

One popular notation and approach to business modeling is the Business Process Modeling

Notation (BPMN) profile (see www.bpmi.org). This notation is specifically targeted at the business modeling community and has a relatively direct mapping to UML through a BPMN Profile. Sparx Systems implements a BPMN Profile and makes available a BPMN Technology add-in for Enterprise Architect for users to extend the basic UML constructs with BPMN notation. A free download of this add-in provides excellent BPMN support with all the graphical icons and behavior specified in the BPMN specification document.

Requirements management

Typically, the first step in developing a solution is to gather requirements, be it to develop a software application or detail a business process. Requirements are essentially ‘what the system needs to do’. Enterprise Architect’s built-in requirements management features can be used to:

- Define an organized and hierarchical Requirements model
- Link and trace the implementation of system requirements to model elements
- Search and report on requirements and perform impact analysis with respect to requirement changes.

Enterprise Architect Editions

Enterprise Architect is available in four editions: Corporate (Floating), Corporate (Standalone), Professional and Desktop. Each edition offers a range of features to support the requirements of different groups of users, from single-person projects to large enterprise teams. The Corporate Floating license arrangement is particularly useful for companies that need to manage a central store of license keys. Floating license keys may be used by different employees over time, on a temporary or permanent basis. More information about Enterprise Architect editions is available here: http://sparxsystems.com/products/ea_editions.html

Process support

UML is a language, not a process. It prescribes the elements of a modeling language and how those elements can be joined together to represent things in the real world. It does not prescribe how you use those elements over time to build new software systems.

Like UML, Enterprise Architect is process neutral, meaning that it includes all the features and functionality necessary to implement some chosen development process, but does not dictate what that process should be or how it should be implemented.

Many Enterprise Architect users adopt highly structured processes, like RUP, while others use more flexible and lighter weight Agile processes. Regardless of the degree of process management you require, Enterprise Architect has the tools and features needed to support the ongoing process of software development.

About Sparx Systems



Sparx Systems is an Australian-based company with a solid history of innovation and development within the modeling/UML market.

Sparx Systems is a contributing member of the Object Management Group (OMG), the standards body responsible for defining and maintaining the UML specification (plus a wide range of other related specifications).

Company vision

Sparx Systems believes that a complete modeling and design tool should be used throughout the full software life-cycle. Our subscription plan reflects this, as does our belief that 'life-cycle' software should be as dynamic and modern as the systems you design and maintain.

Sparx software is intended for use by analysts, designers, architects, developers, testers, project managers and maintenance staff; that is, almost everyone involved in a software development project and in business analysis. It is Sparx Systems' belief that highly priced CASE tools severely limit their usefulness to a team, and ultimately to an organization, by narrowing the effective user base and restricting easy access to the model and the development tool. To this end, Sparx Systems is committed to both maintaining an accessible pricing model and to distributing a 'Read Only' (Enterprise Architect Lite) version of Enterprise Architect for use by those who only need to view modeling information.

Ongoing commitment to Enterprise UML tools

Sparx Systems has been developing enterprise modeling tools for over ten years, and has been active in developing the capabilities of Enterprise Architect to reflect the needs of enterprise modeling. In addition, Sparx has been diligent in listening to requests and suggestions from its extensive user base, implementing many features critical to developing useful and well-rounded models. Sparx Systems is committed to the ongoing development both of their UML modeling tool, Enterprise Architect, and of the numerous plug-in technologies that address domain specific requirements.

Contact Details

Contact Sparx Systems at the following email addresses:

- Sales and Purchase Enquiries: sales@sparxsystems.com
- Product Support Enquiries: support@sparxsystems.com