

Sparx Systems

Enterprise Architect

Reviewer's Guide

Copyright Sparx Systems 2010



Table of Contents

Introduction.....	6
What is Enterprise Architect?.....	6
What differentiates Enterprise Architect from other UML tools?	6
How popular is Enterprise Architect now?	7
What benefits does Enterprise Architect provide?	7
Model and Manage Complex Information.	7
Model, Manage and Trace Requirements.	8
Integrate Teams and Share a Vision.	8
Design and Build Diverse Systems using UML.	8
Visualize, Inspect and Understand Complex Software.....	9
Use Full Lifecycle Modeling and Project Management.	9
Share and Re-Use Information Across Tools.	9
Create Platform Independent Models using Model Driven Architecture.	10
Summary of Enterprise Architect features.....	11
Some Enterprise Architect features in detail	12
UML 2.1 support.....	12
Documentation and reporting in Enterprise Architect.....	12
The Rich Text Format (RTF) report writer	12
The HTML report writer.....	13
Built-in document editor and linked documents	13
Requirements management.....	13
Structured Use Case Scenarios	13
Business Process Modeling.....	13
The BPMN Profile	14
BPEL Generation	14
Model Validation	14

User interface, tools and productivity boosters	14
Model patterns	14
User Interface	14
In-line editing of diagram elements	15
Quick linking technology	15
Other Diagramming features.....	15
Traceability	15
Auditing view	15
Element list	16
Element Browser	16
Relationship Matrix	16
Track element usage.....	16
Traceability Window	16
Model Search.....	16
Model Views	17
Other reports.....	17
Team development and collaboration	17
Support for large models and many concurrent users.....	17
XML import/export	18
Security	18
The Project discussion forum	18
Enterprise Architecture Framework support	18
SOA (Service Oriented Architecture) support	18
XSD Capability.....	19
WSDL Capability.....	19
Code Engineering and visualization	19
Import .jar files and .NET assemblies	20
Template-driven source code generation	20

On-demand and live-generation of code.....	20
Built-in syntax highlighting source code with dynamic source ‘outliner’	20
Visualizing, debugging and profiling executing code.....	20
Built-in support for JUnit and NUnit testing	21
Unit testing integrated into the modeling-development cycle.....	21
Performance and scalability	21
Version Control.....	21
Model Baseline Compare and Merge	22
Connectivity and integration with other tools.....	22
Automation Interface	22
Add-ins.....	23
MDG Link	23
MDG Integration.....	23
MDA (Model Driven Architecture) support.....	23
Database modeling support	24
Built-in data modeling profile.....	24
Support for major database management systems	24
DDL generation	25
Import of database structures from ODBC connections	25
Systems Engineering support	25
SysML and Model Simulation	25
Advanced Behavioral Modeling and executable code generation	25
Hardware Description Languages.....	26
Project Management support	26
Project estimation with Use Case metrics.....	26
Resources.....	26
Testing	26
Change management.....	26

Model tasks.....	27
Glossary	27
Variety of XMI formats	27
Extending Enterprise Architect.....	27
UML Profiles	27
UML Patterns.....	28
MDG Technologies.....	28
Custom Add-ins.....	28
Enterprise Architect editions.....	29
Process support	29
About Sparx Systems	30
Company vision.....	30
Ongoing commitment to Enterprise UML tools	30
Contact details.....	30

Introduction

This document provides a comprehensive overview of the capabilities of Sparx Systems' **Enterprise Architect**. 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 visual platform 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, maintenance, 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: Enterprise Architect makes it easy to get up and running with UML quickly

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?

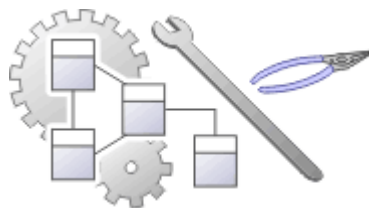
With over 220,000 licenses sold, Enterprise Architect has proven remarkably popular across a wide range of industries and is used by thousands of companies world-wide. From large, well-known, multi-national organizations to smaller independent companies and consultants, Enterprise Architect has become the UML modeling tool of choice for developers, consultants and analysts in over 130 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 1000 seats working on large, distributed projects.

What benefits does Enterprise Architect provide?

Model and Manage Complex Information.

Enterprise Architect helps individuals, groups and large organizations model and manage complex information. Often this relates to software development and IT systems design and deployment, but it can also relate to business analysis and business process modeling. Enterprise Architect integrates and connects a wide range of structural and behavioral information, helping to build a coherent and verifiable architectural model, either what-is or what-will-be. Tools to manage versions, track differences, audit changes and enforce security help control project development and enforce compliance to standards.



Model, Manage and Trace Requirements.

Capture requirements and use full traceability from base requirements to design, build, deployment and beyond. Use impact analysis to trace from proposed changes to original requirements. Build the 'right' system.



Integrate Teams and Share a Vision.

A scalable, easily deployed, multi-user environment, Enterprise Architect integrates team members from all sections and all phases of a product's (or system's) development and maintenance lifecycle, providing significant benefits from the built-in collaboration and inherent information sharing. A single repository for business analysts, software architects, developers, project managers, testers, roll-out and support staff. A 'unified' view of a complex system having many view points and many possible sub-systems.



Design and Build Diverse Systems using UML.

UML 2.1, an open standard, provides a rich language for describing, documenting and designing software, business and IT systems in general. Enterprise Architect allows you to leverage the full expressive power of UML 2.1 to model, design and build diverse systems in an open and well understood manner. Generate code, database structures, documentation and metrics. Transform models. Specify behavior and structure as the basis for contractual agreements.



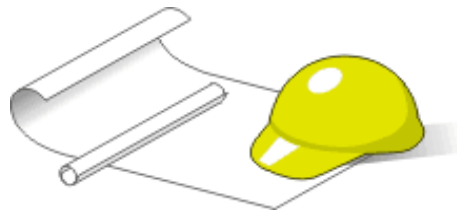
Visualize, Inspect and Understand Complex Software.

Software is complex and often hard to understand. Use Enterprise Architect to reverse engineer a wide variety of source code to understand static structure. To complete the picture, use the unique built-in profiling and debugging tools to capture and visualize executing software at run-time. Reverse engineer database schema for a wide range of systems to integrate existing data models into models. Create run-time instances of model elements and invoke methods using the built in Object Workbench.



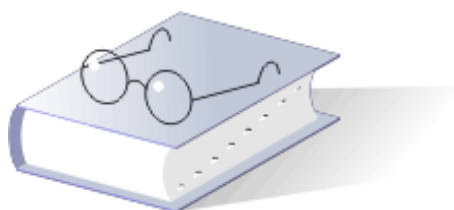
Use Full Lifecycle Modeling and Project Management.

Capture and track information about model elements that are important to success: for example, Testing, Project Management and Maintenance details. Use this information to drive and track product development and delivery.



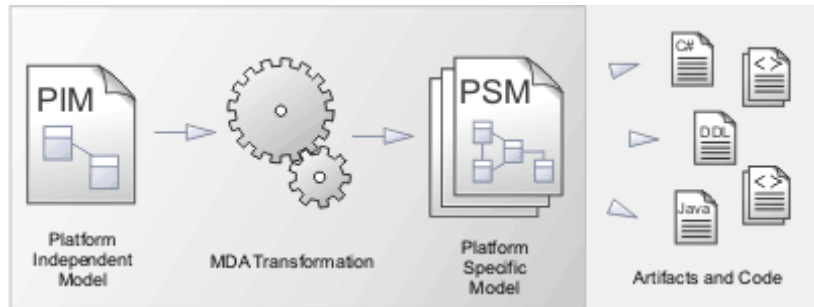
Share and Re-Use Information Across Tools.

Enterprise Architect supports a number of mechanisms for exporting and importing models using industry standard XMI. This allows modelers to use information created in other tools, to copy information between Enterprise Architect models and even to write and use custom tools that take XMI directly as input.



Create Platform Independent Models using Model Driven Architecture.

Model Driven Architecture (MDA) is an open standard designed to facilitate rapid application development in a platform independent manner. Models can be built at a high level of abstraction and using MDA based tools, transformed into models and code targeting a specific platform or domain. Enterprise Architect has a rich set of tools built-in to support MDA.



Summary of Enterprise Architect features

Enterprise Architect enables you to:

- Model complex information, software and hardware systems using UML-compliant notation
- Model, Manage and Trace Requirements to deployed solutions
- Produce detailed and quality documentation in RTF and HTML formats
- Leverage industry-standard Enterprise Architecture Frameworks
- Generate and reverse engineer code in 10+ programming languages¹
- Model databases, generate DDL scripts, and reverse database schema via ODBC*
- Manage, track and control change using baseline model merge and auditing capabilities
- Centralize enterprise-wide documentation of processes and information systems
- Model dependencies between elements, system dynamics and state
- Model class hierarchies, deployment, components and implementation details
- Record project issues, tasks and system glossary
- Assign resources to model elements and track effort expended against required effort
- 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 and Trace relationships between elements using the tabular Relationship Matrix
- Script and automate common tasks using a detailed Automation Interface and Model Scripts
- Connect to shared database repositories using MS SQL Server, MySQL, Oracle and more*
- Migrate changes across a distributed environment using Controlled XMI Packages
- Perform model-to-model transformations using Model Driven Architecture (MDA)*
- Create and share dynamic views of model elements and diagram sets using Model Views
- Create Mind Maps, Business Process Models and Data Flow Diagrams using UML
- Generate BPEL scripts automatically from Business Process models*
- Generate executable business logic from rule tasks and trace to natural language business rules*
- Visualize executing applications using the Debug & Profiling Workbench
- Transform behavioral models into executable source code for software and hardware description languages (HDLs) such as Verilog, VHDL, and SystemC*
- Simulate SysML parametric models*

¹* Available features vary across editions. See <http://www.sparxsystems.com/products/ea/editions.html>

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.html>.

Enterprise Architect provides additional diagram types that extend the core UML diagrams for business process modeling, mind mapping, formal requirements specifications, data-flow diagrams 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.

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 Microsoft® 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.

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.

Structured Use Case Scenarios

Use Case Scenarios capture vital analysis information in the form of natural language descriptions. Enterprise Architect's Structured Scenario editor helps you use this information to drive downstream development and maximize traceability across the development life-cycle.

The Structured Scenario editor helps you to dynamically link scenario steps to associated model elements, such as domain elements, business rules and glossary terms. From structured scenarios, you can automatically generate test case descriptions, Activity and other UML behavior diagrams. You can even reverse engineer existing process diagrams into structured, textual specifications to produce documentation deliverables.

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 provides a built-in UML profile for BPMN modeling in Enterprise Architect.

BPEL Generation

Enterprise Architect supports generating Business Process Execution Language (BPEL) scripts from BPMN models. Generated BPEL scripts can then be executed using a third-party execution engine. Specific model validation rules help to ensure that models will generate valid BPEL code.

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.

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 you can arrange to suit your work methods. Key amongst these windows 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 colored 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

Export diagrams to a range of image formats (.bmp, .jpg, .png, .gif, .emf and .wmf)

Swimlanes enable logical partitioning of diagrams

Pan and Zoom window for easy navigation and preview of complex diagrams

Diagram Filters to dynamically compare 'as-is' and 'to-be' architectures or highlight changes

Lock diagrams to prevent accidental modification

Shape Scripts to customize rendering of elements and support alternative modeling notations

You can also superimpose alternative images on 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 edit 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.

Element Browser

The Element Browser provides a context-sensitive view of all meta-data related to the selected element. This allows you to trace information on testing, project management, structured scenarios, maintenance and model semantics. The Element Browser thus provides a central hub for dynamic model reviews and a launchpad for related actions.

Relationship Matrix

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

Track element usage

Enterprise Architect makes it easy to track and display the use of an element. The 'Find' or '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.

Traceability Window

The Traceability Window provides a dynamic, navigable view of the current element's relationships to other elements in the model. Relationships shown include Aggregation, Inheritance and Dependency; embedded elements are also shown. By highlighting interconnections between levels of abstraction in the model, the Traceability Window provides a powerful impact analysis tool – helping you to see the effect of changing Requirements on downstream elements.

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.

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.

Model Views

Enterprise Architect's Model Views window provides a dynamic, filtered view of elements from the underlying model hierarchy. Using Model Views, you can organize elements according to search criteria (for example frequently run searches), favorite elements and diagrams or technology-specific information, such as elements belonging to a particular framework view point. Model views can be stored locally for use by individuals, imported and exported between users or included in a shared repository to achieve collaborative views. Automatic notifications can be set for a given Model View to alert you when an element created by another author is added to the view.

Other reports

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

- Resource and Task Details
- Project Issues Project
- Glossary Project (size)
- Statistics
- Dependency and Implementation Details
- Testing Details

Team development and collaboration

Enterprise Architect offers specific functionality 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.

Support for large models and many concurrent users

The Corporate Edition of Enterprise Architect enables the use of dedicated DBMS (server based) repositories rather than the standard .EAP files to store shared model data. Enterprise Architect supports the following DBMSs as model repositories:

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

XML import/export

Enterprise Architect supports an XML-based model interchange format known as XML Metadata Interchange (XMI). You can use Enterprise Architect's XMI facility to share model information between developers. XMI enables you to export discrete packages or entire model branches into XML files, which can be imported into other models or maintained in a version control repository.

Security

User security is available in the Enterprise Architect Corporate edition and 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.

The Project 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.

Enterprise Architecture Framework support

Sparx Systems supports industry-standard architectural frameworks to facilitate enterprise modeling projects. Framework implementations in Enterprise Architect are based on UML and its related specifications, which maximizes architectural rigor and allows users to exchange enterprise model information using standards such as XMI. The following architectural frameworks are available as plug-in extensions to Enterprise Architect:

The Zachman Framework (see <http://www.sparxsystems.com/zachman>)

DoDAF (see <http://www.sparxsystems.com/dodaf-modaf>)

MODAF (see <http://www.sparxsystems.com/dodaf-modaf>)

UPDM (see <http://www.sparxsystems.com/updm>)

The Open Group's TOGAF (see <http://www.sparxsystems.com/togaf>)

Including the Federal Enterprise Architecture Framework (FEAF) reference model

SOA (Service Oriented Architecture) support

Enterprise Architect implements the OMG's Service oriented architecture Modeling Language (SoaML), which provides a standards-based approach to modeling SOA solutions using the UML.

As a natural complement to SoaML, Enterprise Architect also supports the [Service Oriented Modeling Framework \(SOMF\)](#), which encourages a holistic view of enterprise software entities via a technology-independent notation. The SOMF modeling notation provides an intuitive approach to

visualize “used-to-be”, “as-is” and “to-be” states of the enterprise service portfolio.

Enterprise Architect also enables you to rapidly model, forward 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, and the coupling of UML 2.1 with XML provides a natural mechanism for specifying, constructing and deploying XML-based SOA artifacts within an organization. Enterprise Architect’s XSD and WSDL capabilities are used to facilitate [BPEL generation](#) from UML models.

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.

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 in the Professional, Corporate and extended suite 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

Import .jar files and .NET assemblies

Enterprise Architect enables you to reverse-engineer the following types of binary modules:

- Java Archive (.jar)

- .Net PE file (.exe, .dll)*

- Intermediate Language file (.il).

** native Windows DLL and Exe files are not supported, only PE files containing .Net assembly data*

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.

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.

Visualizing, debugging and profiling executing code

Enterprise Architect's Visual Execution Analyzer provides facilities to model, develop, debug, profile and manage an application from within the modeling environment. Outputs generated by the Visual Execution Analyzer benefit the development process by:

- Giving you a better understanding of how your system works

- Enabling you to document system features automatically

- Providing information on the sequence of events that lead to erroneous events or unexpected system behavior

Analysis tools provided by the Visual Execution Analyzer can be used to:

- Generate Sequence diagrams, recording live execution of an application, or specific call stacks
- Derive State Transition diagrams, illustrating changes in data structures
- Create Profiler reports, showing application sequences and operation call frequency
- Optimize existing system resources and understand resource allocation
- Ensure that the system is following the rules as designed
- Produce high quality documentation that accurately reflects system behavior
- Understand how and why systems and existing code work
- Train new employees in the structure and function of a system
- Identify costly or unnecessary function calls
- Illustrate interactions, data structures and important relationships within a system
- Trace problems to a specific line of code, system interaction or event
- Visualize why a sequence of events is important
- Establish the sequence of events that occur immediately prior to system failure

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.

Unit testing integrated into the 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 test case transformation on the class. Enterprise Architect creates a corresponding test method that you can generate and fill in the details. This can easily be done before the code you are testing has been created.

Performance and 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.tortoisecvs.org>

Model Baseline Compare and Merge

The Enterprise Architect Corporate edition provides a facility to 'Baseline' (snapshot) a model package at a particular point in time. The baseline can then be used with Enterprise Architect's Compare (diff) utility to visually explore changes to the package at a later point in development. Any differences can be merged from the baseline into the current model, allowing you to 'roll-back' changes to a previous revision of that package. This also allows multiple users to contribute revisions to a given package offline and later incorporate them back into the common model.

As well as comparing and merging changes from a baseline stored within the current model, Enterprise Architect's Compare utility allows you to compare a package against:

- A file on disk, created using the Enterprise Architect XMI export facility on the package

- A version-controlled XMI file for the selected package

- Any baseline of the package residing in an external model to which you have access

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 (MDG) 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

Execute Ad hoc queries against the model.

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® and Eclipse IDEs respectively.

MDG Integration

MDG Integration tightly integrates Enterprise Architect into the Microsoft® Visual Studio® 2005 and 2008 and Eclipse development environments. This product enables users to explore and edit the UML model inside Visual Studio or Eclipse and also provides many of the key features of Enterprise Architect directly within these IDEs, including rich text and web-based document generation, MDA transformations, Baseline management and Engineering of key XML-based technologies.

Other Eclipse-based environments that MDG Integration supports include Adobe® Flex® Builder™ and Progress OpenEdge® Architect.

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's MDA capabilities allow you to:

Leverage built-in transformations to automatically derive:

- Data Models (DDL)
- Code Models, including C# and Java

- XML models, such as XSD and WSDL
- Test Model for JUnit and Nunit
- Activity diagrams and test scripts from Structured Use Case Scenarios

Define new transformations using a powerful, template driven approach

Repeat transformations to ensure consistency between source and target models as they change over time

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, primary keys, foreign keys

Create stored procedures

Create indexes, sequences, functions and triggers

Generate DDL for a table or a package

Convert datatypes for a table, package or entire DBMS

Import a database schema from an ODBC data source

Create views

Support for major database management 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 nine 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.

Systems Engineering support

Integrating many high-end features for Systems Engineers, the Ultimate and Systems Engineering editions of Enterprise Architect provide built-in support for SysML 1.1, parametric model simulation, executable code generation, as well as model to code transformations for Hardware Description Languages and ADA 2005.

SysML and Model Simulation

Enterprise Architect enables you to develop SysML models quickly and simply, through a tightly integrated profile. The SysML profile for Enterprise Architect supports each of the nine SysML 1.1 diagram types, model validation rules and model guidance using the Quick Linker. The Systems Engineering and Ultimate editions of Enterprise Architect also provide simulation of SysML Parametric diagrams. This supports engineering analysis of critical system parameters, including the evaluation of key metrics such as performance, reliability and other physical characteristics.

Advanced Behavioral Modeling and executable code generation

Enterprise Architect can associate detailed behaviors with UML Activity and Interaction elements, allowing for executable code to be generated from Interaction (Sequence) and Activity models. This allows for more than just code stubs to be derived from models, by generating any nested behavioral models that define executable statements, such as conditional logic, looping constructs and assignments.

Hardware Description Languages

The Systems Engineering and Ultimate editions of Enterprise Architect support code generation from State Machine models into Hardware Description Languages (HDLs) such as VHDL, Verilog and SystemC.

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.

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

Enterprise Architect adds a model glossary to each development project, making it simple to define and disseminate new terms and definitions, which may be 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. Enterprise Architect has a generic UML Profile mechanism that allows users to create, import and use profiles as seamless extensions to the core modeling environment. For example, the UML Profile for XML Schema defined by David Carlson in *Modeling XML Applications with XML* is available for Enterprise Architect. This profile describes a set of extensions to basic UML model elements to enable accurate modeling of XSD Schemas Implementations of profiles

UML Patterns

Enterprise Architect's support for UML Patterns provides an excellent means of achieving re-use and robustness. Patterns represent a group of collaborating objects and classes that can be abstracted from a general set of modeling scenarios. 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

MDG Technologies encapsulate a logical collection of resources (such as UML patterns and profiles) that pertain to a specific technology or modeling domain. These are 'pluggable' resources for Enterprise Architect that reside either in a physical directory or URL.

Users may create their own MDG Technologies, or take advantage of technologies built into the Enterprise Architect Installer, such as ICONIX and Mind Mapping. Free plug-in technologies that support such activities as CORBA code engineering and BPMN model validation are available for download from: http://www.sparxsystems.com/resources/mdg_tech. In addition, licensed solutions for systems modeling (SysML), Distributed Data Services (DDS) and others are also available – see <http://www.sparxsystems.com/products/#MDGTech>.

Custom Add-ins

Add-ins 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.

Enterprise Architect editions

Enterprise Architect is available in six editions: Ultimate, Systems Engineering, Business and Software Engineering, Corporate, 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.



A floating license arrangement is also available for Ultimate, Systems Engineering, Business and Software Engineering and Corporate editions. The floating license 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 from:

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 and 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 model 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 its UML modeling tool, Enterprise Architect, and of the numerous plug-in technologies that address domain specific requirements.

Contact details

Company Address: Ferrogasse 80/3, 1180 Vienna, Austria

Office: Handelskai 340, 1020 Vienna, Austria

Phone.: +43 (0)662 90 600 2041

Fax: +43 (0)662 90 333 3041

e-Mail: sales@sparxsystems.eu

Order Processing: sales@sparxsystems.eu

Product Support: support@sparxsystems.eu

Internet German: www.sparxsystems.de

Internet English: www.sparsystems.eu

Blog: blog.sparxsystems.at