

Sparx Systems

Enterprise Architect 9

im Überblick

©Copyright SparxSystems GmbH 2011



www.sparxsystems.eu

Inhalt

<i>Einleitung</i>	4
<i>Was ist Enterprise Architect?</i>	4
<i>Was unterscheidet Enterprise Architect von anderen UML Werkzeugen?</i>	4
<i>Wie verbreitet ist Enterprise Architect heute?</i>	5
<i>Welche Vorteile bietet Enterprise Architect?</i>	5
<i>Enterprise Architect – Funktionen im Überblick</i>	8
<i>Einige Enterprise Architect Features im Detail</i>	9
UML 2.1 Unterstützung	9
Dokumentations- und Reporterstellung im Enterprise Architect	10
<i>Der Rich Text Format (RTF) Template-/Reporteditor</i>	10
<i>HTML Reporterstellung</i>	10
<i>Der eingebaute Dokumenteneditor und das Verlinken von Dokumenten</i>	10
Requirements Management und Anforderungsmanagement	10
Unterstützung von Szenarios/Storyboards	11
Modellierung von Geschäftsprozessen.....	11
<i>Das BPMN Profil</i>	11
Modell-Validierung	11
Benutzerinterface, Werkzeuge und Produktivitätsanreicherung	12
<i>Modellvorlagen</i>	12
<i>Benutzer-Interface</i>	12
<i>Direktes Editieren von Diagrammelementen</i>	12
<i>“Quick Linking” Technik</i>	12
<i>Weitere Diagramm-Funktionen</i>	12
Nachvollziehbarkeit	13
<i>Protokollierung (Auditing)</i>	13
<i>Elementliste</i>	13
<i>Nachvollziehbarkeit der Elementverwendung</i>	13
<i>Die hierarchische (Abhängigkeits-)Ansicht</i>	14
<i>Die Matrixansicht</i>	14
<i>Diagrammerstellung mittels Elementeverkettung</i>	14
<i>Modellsuche</i>	14
<i>Modellsichten</i>	14
<i>Andere Berichte</i>	14
Entwicklung und Zusammenarbeit im Team	15
<i>Unterstützung großer Modelle oder vieler gleichzeitiger Anwender</i>	15
XML Import/Export	15
Sicherheit/Locking	15
Das Modell-Diskussionsforum.....	16
Unterstützung von Frameworks für die Unternehmensarchitektur	16
Unterstützung von SOA (Service Oriented Architecture).....	16
<i>XSD Funktionalität</i>	16
<i>WSDL Funktionalität</i>	16
Codeumsetzung und Codedarstellung	16
<i>Import von .jar files und .NET assemblies</i>	17
<i>Vorlagengetriebene Codeerstellung</i>	18
<i>Angestoßene und automatische “live”-Codeerstellung</i>	18
<i>Eingebauter Codeeditor mit Hervorhebungen und dynamischer Strukturanzeige</i>	18
Debugging, Compiling und Darstellung von Programmen.....	18
<i>Erstellen und Ausführen von Scripts</i>	19

<i>Kompilieren und Ausführen mit der Debug Workbench</i>	19
<i>Automatische Erzeugung von Sequenzdiagrammen</i>	19
Eingebaute Unterstützung für JUnit und NUnit Tests	19
<i>Integration des Unit-Testings in den Entwicklungszyklus</i>	20
Performanz und Skalierbarkeit	20
Versionierung	20
<i>Baselines und Differenzenermittlung</i>	20
Zusammenspiel und Integration mit anderen Werkzeugen.....	21
Automation Interface.....	21
Add-ins	21
<i>MDG Link</i>	21
<i>MDG Integration</i>	21
MDA (Model Driven Architecture) Unterstützung.....	22
Datenbankmodellierung	22
<i>Verfügbares Datenbank-Modellierungsprofil</i>	22
<i>Unterstützung gängiger Datenbanksysteme</i>	23
<i>DDL Erstellung (Eingangsdatei für den Schema-Prozessor)</i>	23
<i>Import von Datenbankstrukturen aus ODBC-Verbindungen</i>	23
Unterstützung der Projektabwicklung	24
<i>Projektkennzahlenermittlung mittels "Use Case Metrics"</i>	24
<i>Ressourcen</i>	24
<i>Testen</i>	24
<i>Change Management</i>	25
<i>Projektaufgaben</i>	25
<i>Wörterbuch</i>	25
<i>Präsentationsfunktionen</i>	25
Import - Export Funktionalitäten.....	25
<i>Unterschiedliche XMI Formate</i>	25
<i>CSV</i>	26
UML-Erweiterungen in Enterprise Architect.....	26
<i>UML Profile</i>	26
<i>UML Muster (Patterns)</i>	26
<i>MDG Technologie</i>	26
<i>Spezielle Add-Ins</i>	27
<i>Enterprise Architekt Editionen</i>	27
<i>Prozessunterstützung</i>	27
<i>Über Sparx Systems</i>	28
SparxSystems Central Europe Software GmbH ist seit Mai 2004 in Europa aktiv, wobei zunächst mit der Übersetzung für den deutschsprachigen Markt begonnen wurde. Nunmehr haben wir uns als lokaler Ansprechpartner für den gesamten deutschsprachigen Raum etabliert.	28
Firmen-Vision	28
Langfristige Ausrichtung auf unternehmensweite UML Werkzeuge	28
Kontaktdaten	28

Einleitung

Dieses Dokument gibt einen umfassenden Überblick über die Funktionalität und die Leistungsfähigkeit von Sparx Systems' **Enterprise Architect 9 (EA)**. Jedem speziellen Merkmal wird ein Abschnitt gewidmet - jeweils mit Erläuterung des Zwecks und des Nutzens der Funktion.

Was ist Enterprise Architect?



Enterprise Architect ist ein grafisches Computer Aided Software Engineering (CASE) Tool, also ein Softwareentwicklungswerkzeug zum Entwurf und zur Herstellung von Softwaresystemen, zur Geschäftsprozessmodellierung und zur Modellierung beliebiger Prozesse oder Systeme.

Enterprise Architect stützt sich auf die letztgültige UML 2.3 Spezifikation (Unified Modeling Language, siehe www.omg.org). UML definiert eine grafische Symbolsprache, um ein System zu beschreiben – während der Planungsphase oder zur Dokumentation eines bestehenden Systems.

Enterprise Architect ist ein fortschrittliches Werkzeug, das **alle Teile des Entwicklungszyklus** abdeckt und eine umfassende Nachvollziehbarkeit des Projekts vom ersten Entwurf bis zur Auslieferung und bis in die Wartungsphase hinein zulässt. Requirements Management, die Testphase und auch das Änderungsmanagement werden unterstützt.

Was unterscheidet Enterprise Architect von anderen UML Werkzeugen?

- Die umfassende Unterstützung von UML 2.3
- Die Möglichkeit, Anforderungen vollständig zu managen (*Requirements Management*).
- Eine umfassende Projektmanagementunterstützung, einschließlich Ressourcenplanung, Aufgaben, Projektkalender, Metriken, Testvorgaben und Testverfolgung.
- Eine Testunterstützung: Testfälle, JUnit und *NUnit* werden unterstützt.
- Flexible Dokumentationsfunktionen: HTML und Rich-Text (RTF) Reportgeneratoren.
- Codeerstellungsunterstützung für zahlreiche Programmiersprachen 'out of the box'.
- Eine integrierte Debuggingfunktion für ausführbare Java und .Net Projekte/Programme, mit automatischer Erstellung eines Laufzeitmodells und eines Sequenzdiagramms aus dem jeweiligen Debuggingprotokoll.
- Eine erweiterbare Modellierungsumgebung mit der Möglichkeit, benutzerdefinierte Profile und Technologien einzubinden.
- Die leichte Bedienbarkeit.
- Geschwindigkeit: EA arbeitet spektakulär schnell.
- Skalierbarkeit: EA kann problemlos extrem große Modelle verarbeiten und unterstützt das parallele Arbeiten mehrerer Benutzer an einem Projekt.
- Kosten: Die Lizenzpreise von EA sind ausgelegt, um das ganze Team ausstatten zu können. Interaktive Zusammenarbeit und eine Entwicklung im Team werden dadurch Realität.

Wie verbreitet ist Enterprise Architect heute?

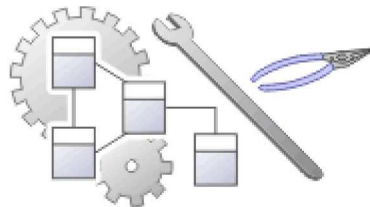
Mit mehr als 250.000 aktiven Lizenzen hat Enterprise Architect seine Popularität in vielen Industriesparten bewiesen und wird von tausenden Firmen weltweit genutzt. Von großen, bekannten, multinationalen Organisationen bis zu unabhängigen Kleinfirmen und Beratern ist Enterprise Architect das UML-Modellierungswerkzeug der ersten Wahl für Entwickler, Konsulenten und Analysten in über 60 Ländern geworden.

Enterprise Architect von SparxSystems wird zur Entwicklung verschiedenster Softwaresysteme in einem breiten industriellen Spektrum verwendet: Luft-/Raumfahrt, Bankwesen, Internetentwicklung, Fertigung, Finanzwesen, Medizin, Militär, Forschung, Lehre, Transport, Einzelhandel, Energieversorgung und Elektrotechnik. EA wird auch weltweit in vielen prominenten Schulen, Universitäten und Schulungsunternehmen effektiv zur UML-Schulung und zur Businessprozessmodellierung eingesetzt. Die Einsatzbreite reicht von einzelnen Anwendern bis zu Großfirmen mit über 600 Nutzern, die an großen, verteilten Projekten arbeiten.

Welche Vorteile bietet Enterprise Architect?

Modellieren und Managen von komplexen Informationen.

Enterprise Architect unterstützt einzelne Personen und Gruppen bis hin zu großen Organisationen beim Modellieren und Managen von komplexen Informationssystemen. Meistens handelt es sich dabei um Softwareentwicklungen, IT-Systeme und deren Design und Entwicklung. Darüber hinaus kann Enterprise Architect zum Modellieren von Unternehmensstrukturen und Geschäftsprozessen verwendet werden. Enterprise Architect integriert und verbindet umfangreiche Struktur- und Verhaltensinformationen um ein kohärentes und überprüfbares Modell eines Soll- oder Istzustandes zu erstellen. Weitere Features sind Funktionen zum Managen von Modellversionen, das Verfolgen von Modelldifferenzen, das Konfigurieren von Zugriffsrechten und vieles mehr. Mithilfe dieser Konfigurationsmöglichkeiten unterstützt Enterprise Architect das strukturierte Erstellen und Abwickeln von Projekten.



Modellieren, Managen und Verfolgen von Anforderungen.

Enterprise Architect unterstützt die Erfassung von Basisanforderungen und deren nachvollziehbare Zuordnung zu Designanforderungen sowie Verteilungsanforderungen. Vorgeschlagene Anforderungen können analysiert, überprüft und akzeptiert werden. Dies ermöglicht die Nachvollziehbarkeit aller Anforderungen von deren Erstellung bis hin zur Realisierung des gewünschten Systems.



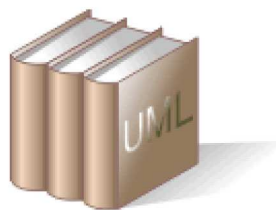
Zusammenarbeit im Team für eine gemeinsame Projektsicht.

Eine skalierbare, einfach zu erstellende Multiuserumgebung erlaubt in Enterprise Architect Teammitgliedern aus verschiedenen Bereichen und Phasen eines (System-) Produktentwicklungs- und Wartungslebenszyklus den Zugriff auf für sie bestimmte Informationen. Die Architektur von EA-Projekten erlaubt es, ein gemeinsames Repository für Businessanalytiker, Softwarearchitekten, Entwickler, Projektmanager, Tester, Roll-Out- und Supportmitarbeiter zu verwenden. Dadurch werden Sichten aus verschiedenen Blickwinkeln auf ein komplexes System und dessen Subsysteme möglich, ohne Inhalte mehrfach anlegen und führen zu müssen.



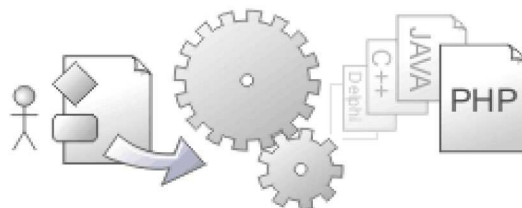
UML basiertes Entwerfen und Entwickeln von Systemen.

UML 2.3 ist ein offener und umfassender Standard, der es erlaubt, Softwaresysteme, Business und IT Systeme zu entwerfen und zu dokumentieren. Durch eine interaktive Führung ermöglicht Enterprise Architect auch dem UML-Einsteiger die Mächtigkeit von UML 2.3 rasch und einfach einzusetzen. Mithilfe der erstellten Modelle können Code, Datenbanken, Dokumentationen und Metriken erstellt werden, ebenso Transformation von Modellen und Struktur- und Verhaltensspezifikationen als Grundlage für vertragliche Vereinbarungen.



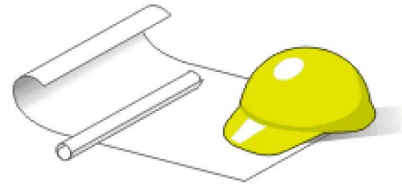
Visualisieren, Überprüfen und Verstehen komplexer Software.

Softwaresysteme sind meist komplex und schwer zu verstehen. Verwenden Sie Enterprise Architect, um bestehenden Source Code reverse zu engineeren - um die Struktur der Implementierung zu verstehen. Um das bestehende Bild eines vorhandenen Systems zu vervollständigen, können die in Enterprise Architect eingebauten Debugging- und Aufzeichnungsfeatures verwendet werden. Damit können Laufzeitmodelle visualisiert werden. Die Datenmodelle bestehender Datenbanken verschiedener DBMS können ebenfalls reverse engineered werden, um sie als visuelles Modell zu repräsentieren. Darüber hinaus können Laufzeitinstanzen eines Modells erstellt werden, um sie mittels der Object Workbench in Enterprise Architect aufrufen zu können.



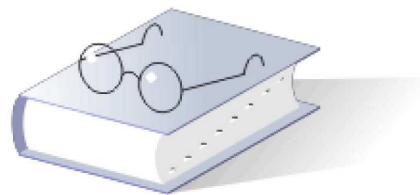
EA unterstützt die Modellierung eines Systemlebenszyklus und das Projektmanagement.

Das Erfassen und Verfolgen von erfolgskritischen Informationen über Modellelemente ist vorgesehen, zum Beispiel: Teststatus, Projektmanagement- und Wartungsdetails. Diese Informationen werden verwendet, um den aktuellen Entwicklungsprozess und den Projektzustand zu verfolgen.



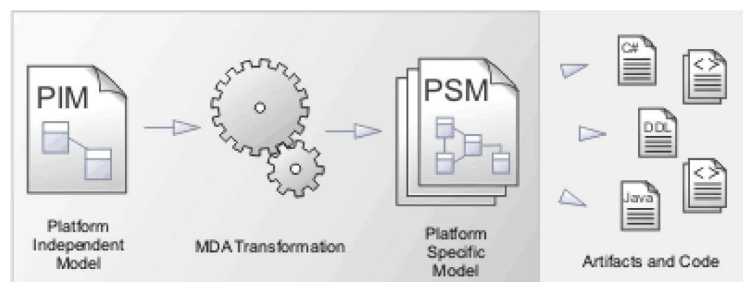
Verteilen und Benützen von Informationen über Toolgrenzen hinaus.

Enterprise Architect unterstützt mehrere Mechanismen für Import und Export von Modellen. Durch diese Mechanismen können Informationen aus anderen Tools benützt werden, sowie Informationen zwischen Enterprise Architect Modellen und Modellen anderer Modellierungswerkzeuge ausgetauscht werden. EA unterstützt mehrere XMI-Standards (www.omg.org) und csv. Der Modellinhalt im Repository kann auch über ein im EA eingebautes Scripting-Tool (VB-, J- und Java-Script) gelesen und verändert werden. Sie können aber auch von außen aus jeder active-x/ole-fähigen Programmiersprache lesend und schreibend zugreifen – eine ausführlich Dokumentation unterstützt Sie dabei.



Erstellung von plattformunabhängigen Modellen durch modellgetriebene Softwareentwicklung.

Modellgetriebene Softwareentwicklung (*Model Driven Architecture*) (MDA) ist ein offener Standard um konzeptuelle und plattformunabhängige Modelle (*Plattform Independent Models*) (PIM) mit Hilfe von Transformationen in verschiedene plattformspezifische Modelle (*Plattform Specific Models*) (PSM) zu übersetzen. Aus den erstellten plattformspezifischen Modellen kann Code generiert werden. Enterprise Architect beinhaltet mehrere vordefinierte Transformationen, um konzeptuelle Modelle in plattformspezifische Modelle zu übersetzen. Diese Transformationen werden über Templates gesteuert, die der Anwender auch verändern kann. Ebenso können Sie weitere Transformationen mittels selbst erstellter, neuer Templates hinzufügen.



Enterprise Architect – Funktionen im Überblick

EA ermöglicht:

- die Modellierung komplexer Software- und Hardwaresysteme in UML-gerechter Darstellung.
- die Modellierung und das Verwalten von Requirements und die Nachvollziehbarkeit deren Umsetzung im modellierten System – bis in den Code hinein.
- die Erstellung detaillierter und hochqualitativer Dokumentation in den Formaten RTF, PDF und HTML.
- die Erzeugung und die Rückeinbindung (*Reverse Engineering*) von mehr als 10 Programmiersprachen; die bekanntesten sind: Action Script, Ada, C, C++, C#, Delphi, Java, PHP, Python, System C, Verilog, Visual Basic, VHDL und VB.NET
- die Modellierung von Datenbanken und das automatische Erstellen von DDL Scripts, sowie die Rückeinbindung (*Reverse Engineering*) von Datenbankschemata direkt über ODBC Verbindungen
- die Versionierung eines Projektes (intern mittels *Baselines* oder durch Anbindung von VCS⁴) und das Aufzeichnen eines Änderungsprotokolls (*Audit*). Die Versionierung erlaubt Differenzbildung und das Wiederherstellen älterer Versionen, auch wahlfrei auszugsweise.
- den zentralen und unternehmensweiten Zugriff auf modellierte Informationen.
- die Modellierung von Abhängigkeiten zwischen Modellelementen, wie z. B. das Verbinden von dynamischen Modellen mit statischen Modellen .
- das Modellieren von Implementierungsdetails, wie Klassenhierarchien, deren Zusammenfassung zu Komponenten und die physische Verteilung dieser Komponenten.
- die Verwaltung von Projektvorfällen (*issues*), Aufgaben (*tasks*) sowie die Verwendung eines projektspezifischen Wörterbuchs (*glossary*).
- eine Zuordnung von Ressourcen zu Modellelementen und die Erfassung und Gegenüberstellung von tatsächlichem und geplantem Aufwand.
- eine Ausgabe der Modelle in Formaten kompatibel zu XMI 1.0, XMI 1.1, XMI 1.2, XMI 2.0 und XMI 2.1 zum Import in andere XMI-kompatible Werkzeuge.
- eine Übernahme von Modellen in den Formaten XMI 1.0, XMI 1.1, XMI 1.2, XMI 2.0 und XMI 2.1 von anderen Werkzeugen. Weiters werden EMX und UML2 als Importformat unterstützt.
- eine Versionierung über XMI unter Benutzung von SCC, CVS oder Subversion
- die Benutzung von *UML Profilen*, um kundenspezifische Erweiterungen von UML bereitzustellen. Hierdurch können auch weitere Normen, z. B. die eEPK emuliert werden.
- komplette Diagramme als *UML Pattern* zu speichern und diese Pattern im selben oder auch in anderen Projekten wiederzuverwenden.
- die Analyse von Beziehungen zwischen Elementen in Tabellenform mittels der sogenannten Beziehungs-Matrix (*relationship-matrix*).
- Scripts zu schreiben um wiederkehrende Aufgaben durch ein *Automation Interface* (API Schnittstelle zu Enterprise Architect) zu automatisieren.
- das Verbinden zu einem datenbankbasiertem EA-Repository. EA unterstützt eine Reihe von DBMSen als Projektrepository, wie z. B.: SQL Server, MySQL, Oracle9i, PostgreSQL, Adaptive Server Anywhere, und Progress OpenEdge

- die Verteilung von Modelländerungen basierend auf versionskontrollierten Paketen.
- die Erstellung von Modell-2-Modell Transformationen mittels *Model Driven Architect* (MDA) Technologien.
- das Erstellen und Verteilen von dynamischen Sichten auf EA Projekte (Elemente und Diagramme) mittels *Model Views*.
- das Erstellen von *Mind Maps*, *Business Process Models* und *Data Flow* Diagrammen.
- Die automatische Genierierung von BPEL Skripten ais Geschäftsprozessen in BPMN Notation.
- Generierung ausführbarer Geschäftsprozesslogik von Rule Tasks und Nachvollziehbarkeit zu natülichen Geschäftsprozellregeln
- die Möglichkeit, vorhandenen Code in Enterprise Architect auszuführen und zu debuggen.

Einige Enterprise Architect Features im Detail

Jeder der nachfolgenden Abschnitte behandelt einen Teilaspekt der Fähigkeiten von Enterprise Architect und zeigt den Zweck und Nutzen der Funktion auf.

UML 2.3 Unterstützung

Enterprise Architect unterstützt alle in UML 2.3 definierten Modelle und Diagrammartent. Damit können Sie Geschäftsprozesse, Webseiten, Benutzerschnittstellen, Netzwerke, Hardware-konfigurationen, Nachrichten und viele andere Aspekte Ihrer Entwicklung modellieren. Sie können den Aufwand Ihrer Projektarbeit in Stunden planen. Sie können Anforderungen, Ressourcen, Testpläne, Störungen und Änderungsanträgen erfassen und verfolgen, vom ersten Konzept bis zur Produktivsetzung mit Wartung und Produktionsunterstützung. Enterprise Architect hat alle Funktionalitäten, die Sie für Ihren Entwurf und das Management Ihrer Umsetzung und Implementierung brauchen.

In Kurzfassung, EA

- war das erste UML-Werkzeug, das eine Abdeckung des UML 2 Standards implementierte - im April 2004
- wird fortlaufend an die UML 2 Erweiterungen angepasst
- unterstützt alle 14 Diagrammartent von UML 2

Leser, die mit UML nicht vertraut sind, finden eine Kurzeinführung unter folgender Internetadresse: http://www.sparxsystems.com/UML_Tutorial.htm

Enterprise Architect unterstützt folgende Modelle, welche den Kern von UML erweitern: Geschäftsprozessmodelle, Mind Mapping, Requirementsspezifikation, Datenflussmodelle und weitere domänenspezifische Modelle. Die Modellierungsumgebung von Enterprise Architect unterstützt weitere Sichten, welche das Editieren von UML Modellen erleichtern und damit effizienter macht. Ein konkretes Beispiel ist die Zustandstabelle (*state table*), welche UML Zustandsdiagramme als editierbare Tabelle mit Zuständen und Zustandsübergängen darstellt.

Dokumentations- und Reporterstellung im Enterprise Architect

Die Dokumentationserstellung trägt wesentlich zur Wertschöpfung der Verwendung von Enterprise Architect bei. Der EA erzeugt hochqualitative Dokumentation, entweder im RTF-, PDF- oder im HTML-Format. Sie können die RTF-Ausgabe mittels RTF-Vorlagen (*templates*) selbst gestalten. Mit der Funktion „Virtual Documents“ des Enterprise Architects können Sie umfassende Dokumente mit unterschiedlich strukturierten Teilabschnitten in einem Durchgang rasch erstellen – inklusive durchgängiger, korrekter Kapitelnummerierung und eines automatisch erstellten Inhaltsverzeichnisses.

Es gibt viele Möglichkeiten, den Inhalt der RTF-Dokumente zu bestimmen. Sie können:

- durch Auswahl eines Package und/oder seiner Sub-Packages die Dokumentation auf diese Packages einschränken.
- einzelne Sub-Packages aus der Dokumentation ausschließen falls diese und deren Sub-Packages nicht automatisch (rekursiv) mit dokumentiert werden sollen.
- einem Paket oder mehreren Paketen ein RTF-Template zuordnen, um den gewünschten Drucksatz vorzugeben.

Sie können einzelne Packages auswählen, gruppieren und in beliebiger Art und Weise zusammenbinden, auch unabhängig von der Sichtweise im Projekt, bzw. unabhängig von der Gliederung im *Project Browser*. Generierte Dokumente können als PDF erstellt werden, oder auch speziell für Open Office optimiert werden.

Der Rich Text Format (RTF) Template-/Reporteditor

Der RTF-Textvorlagen-Editor erlaubt die Erstellung und das Editieren von anwenderspezifischen Vorlagen zur Erstellung beliebiger RTF-Dokumentationen. Der Vorlageneditor erlaubt die Auswahl beliebiger Arten von Modellelementen und die Auswahl der Feldinhalte des Elements, die in die Dokumentation einzubinden sind. Selektive Kontextmenüs sind zur Vereinfachung der Vorlagenerstellung vorgesehen.

Beliebige RTF-Formatierungen können im Vorlageneditor erstellt werden, ebenso können Tabellen sowie Kopf- und Fußzeilen in das RTF-Template eingefügt werden. Grafiken können eingebunden werden, ole-Verknüpfung zu externen Quellen aus *Linked Documents* ist möglich.

HTML Reporterstellung

Enterprise Architect erlaubt Ihnen, das ganze Modell oder einen einzelnen Zweig des *Project Browsers* als HTML-Struktur zu exportieren. Der HTML-Report ist eine einfach zu benutzende und umfassende Darstellung des Modellbaums – in Übereinstimmung mit der Gliederung im *Projekt Browser*. Die erzeugte HTML-Darstellung ist eine eins zu eins Abbildung des EA Projektes, alle vorhandenen Links zu anderen Elementen und Diagrammen werden auch in der HTML-Darstellung erzeugt. Die HTML-Erstellung benutzt eine durch den Benutzer veränderbare HTML-Vorlage. Damit können eigene Styles erstellt werden, die Ihren Firmenstandards entsprechen.

Der eingebaute Dokumenteneditor und das Verlinken von Dokumenten

Enterprise Architect ermöglicht die Verlinkung von Rich-Text-Dokumenten zu jedem einzelnen Modellelement. Verlinkte Dokumente (*linked documents*) können bei der Generierung von HTML und RTF Dokumenten automatisch mit ausgegeben werden.

Requirements Management und Anforderungsmanagement

Typischerweise ist der erste Schritt der Lösungsfindung die Sammlung von Anforderungen, sei es eine Softwareentwicklung oder eine Geschäftsprozessoptimierung. Anforderungen stellen dar, was

ein System leisten soll. Die im EA eingebaute Anforderungsverwaltung kann verwendet werden zur:

- Anlage eines gegliederten und hierarchisch organisierten Anforderungsmodells.
- Verknüpfung der Systemanforderungen mit Modellelementen und Nachverfolgung der Umsetzung über die Verknüpfungen.
- Suche nach Anforderungen und der Reporterstellung für Anforderungen als auch für eine Auswirkungsanalyse bei Anforderungsänderungen.

Unterstützung von Szenarios/Storyboards

Enterprise Architect unterstützt die Verwendung von Szenarien alias Storyboards. Neben der Möglichkeit, rein textlich alternatives Verhalten zu beschreiben, sind umfangreiche Komfortfunktionen eingebaut.

Aus Texten im EA kann grundsätzlich auf Modellelemente, ganze Diagramme oder Testfälle verlinkt werden, wodurch die logische Aufbereitung eines Anforderungstextes wesentlich erleichtert wird.

Ferner besteht die Möglichkeit, Szenarios als *structured scenarios* zu erfassen, wobei Verzweigungspunkte zu und Zusammenführungen von alternativem Verhalten festlegbar sind. Diese dezimal klassifizierten Szenarios dürfen in ihren Punkten auch um Auslöser, Bedingungen und Zustandsbenennungen ergänzt werden.

Die *structured scenarios* können durch den Enterprise Architect automatisch in Verhaltensdiagramme (Action, State, Sequenz) umgebrochen werden. Ein automatischer Umbruch in der Gegenrichtung, ausgehend von Aktivitätsdiagrammen in *structured scenarios* ist auch enthalten.

Modellierung von Geschäftsprozessen

Es gibt viele Annäherungen, mit UML als Basismodellierungssprache Geschäftsprozesse (business processes modeling BPM) zu modellieren.

Aktivitätendiagramme, Objektdiagramme und die Verwendung von UML-Erweiterungen (mit Profilen) bilden die Möglichkeit zum Modellieren von Geschäftsprozessen. Enterprise Architect ergänzt die Basis-UML-2.3-Sprachpalette durch Elemente für Vorgangsanalyse, Requirements Management (Anforderungsmanagement) und Prozessmanagement, wie z. B. Änderung (*change*), Erweiterung (*feature*) und Projektsachverhalte (*issue*)-Elemente.

Das BPMN Profil

Eine populäre Notation der Geschäftsprozessmodellierung ist die *Business Process Modeling Notation* (BPMN) (siehe auch www.bpmi.org). BPMN ist eine eigene Sprache neben UML und ist der UML Notation und Arbeitsweise sehr ähnlich. Sparx Systems implementiert ein BPMN-Profil und stellt eine BPMN-Technologie als Add-In für Enterprise Architect kostenlos zur Verfügung. Damit erweitern Sie Enterprise Architect um eine mächtige standardisierte Prozessbeschreibungssprache.

Die BPEL-Generierung

Enterprise Architect unterstützt die Generierung von BPEL Skripten (Business Process Execution Language) aus BPMN Modellen heraus. So erstellte BPEL Skripte können mit Hilfe externer Ausführungstools ausgeführt werden. Spezifische Modellvalidierungsregeln helfen ausführbaren BPEL Code generieren zu können.

Modell-Validierung

Die Modellvalidierungsfunktion überprüft UML Modelle auf die Einhaltung definierter UML Regeln als auch alle im Modell festgelegten Beschränkungen unter Verwendung der "Object

Constraint Language“ (OCL). Sie können die eingebaute Modellvalidierung auf ein einzelnes UML-Element, ein Diagramm oder ein ganzes Package anwenden.

Benutzerinterface, Werkzeuge und Produktivitätsanreicherung

Modellvorlagen

Der Enterprise Architect enthält bereits eine große Anzahl an Mustermustern, die zur Erstellung neuer Projekte und Modelle verwendet werden können. Jede Vorlage enthält hilfreiche Anmerkungen, Referenzen und Ausgangselemente. Alle gängigen UML Design-Pattern stehen als Vorlage im EA zur Verfügung. Darüber hinaus können Sie mit des Enterprise Architects eigene Pattern erstellen und in neuen Projekten wiederverwenden.

Benutzer-Interface

Die Benutzeroberfläche von Enterprise Architekt besteht aus einer Reihe von Menü-Fenstern, Menübalken und Werkzeugkästen, die vollkommen flexibel ein- und ausgeblendet werden können und an beliebigen Bildschirmbereichen positioniert werden können. Folgende Aufzählung beschreibt die wichtigsten Benutzerinterfaces:

- Der *Projekt Browser* ist die Ablage für alle Modellelemente eines Projektes. Jedes Modellelement ist in einer Paketstruktur organisiert, welche eine Projekthierarchie bildet (optional können Ebenennummern eingeblendet werden). Der Projekt Browser erlaubt, Modellelemente und Diagramme des gesamten Projektes zu selektieren, zu editieren, zu reorganisieren und zu löschen.
- Der Enterprise Architekt Werkzeugkasten (*Toolbox*) ist kontextsensitiv und zeigt abhängig vom geöffneten Diagrammtyp die passenden Elemente und Konnektoren an. Der Werkzeugkasten ermöglicht ein schnelles und bequemes Auswählen von Elementen und Verbindungen zwischen Elementen, egal ob es sich dabei um UML-Elemente oder um erweiterte Sprachelemente handelt, die mittels selbst definierbaren *Technologien* erstellt werden können.
- Das Diagrammfenster erlaubt, Diagramme durch Auswahl von Elementen aus dem Projekt Browser zu erstellen. Die Farben von Diagrammen und Elementen können für eine bessere Differenzierung verändert werden. Dabei kann zwischen einfarbigen Hintergrundfarben und Farben mit Farbverlauf gewählt werden.
- Das Kontextmenü erlaubt den schnellen Zugriff auf Aktionen, die für das jeweils selektierte Element zur Verfügung stehen.

Direktes Editieren von Diagrammelementen

Enterprise Architect erlaubt ein rasches Editieren von Elementeigenschaften direkt aus dem Diagramm. Durch Nutzung der Direkteingabekommandos können Sie rasch und einfach Elementeigenschaften wie Attribute, Methoden und Parameter eingeben, ohne das Diagramm verlassen zu müssen. Oft benötigte Fenster, wie z. B. die Fenster für Attribute und Methoden sind amodal gehalten, sodass Sie rasch zwischen Klassen wechseln können, ohne die Fenster schließen und wieder öffnen zu müssen.

“Quick Linking” Technik

Der “Quick Linker” stellt einen Mechanismus bereit, um rasch neue Elemente und Verbindungen in einem Diagramm einzufügen. Er bietet kontextabhängige Menüs an, die es erleichtern, grammatikalisch korrekte Modelle rasch, auch ohne umfangreiche UML-Kenntnisse, zu zeichnen.

Weitere Diagramm-Funktionen

- Sie können Diagramme in einer großen Anzahl verschiedener Formate exportieren (.bmp,

.jpg, .png, .gif, .emf and .wmf).

- Swimlanes erlauben die logische Aufteilung von Diagrammen.
- Zoom-Fenster zur raschen Vorschau und Navigation komplexer Diagramme.
- Diagrammfilter zum dynamischen Vergleich von Ist- und Sollarchitekturen oder der Darstellungen von Änderungen.
- Sperren von Diagrammen um versehentliche Änderungen zu verhindern.
- Mit Enterprise Architect können Sie die UML Notationselemente in Verbindung mit *Stereotypen* verändern. Für ein bestimmtes *Stereotyp* eines Notationselements kann ein *shape-script* erstellt und zugeordnet werden. Wird dem Element das Stereotyp zugeordnet, wechselt die Darstellung im Diagramm automatisch auf den definierten Shape.
- Ähnlich wie mit *shape-scripts* die UML-Notation verändert wird, kann durch das Laden von alternativen Bildern eine vorhandene Grafik oder der Clipboardinhalt anstelle des UML-Notationssymbols angezeigt werden. Dies wird oft bei UML-Verteilungsdiagrammen angewandt.

Nachvollziehbarkeit

Protokollierung (Auditing)

Die Protokollfunktion des Enterprise Architects erlaubt das Aufzeichnen von Modellveränderungen. Bei eingeschaltetem Protokoll (*audit*), kann der dazu berechtigte Anwender folgende Informationen eruieren:

- Wer veränderte ein Element?
- Wie viele Elemente wurden von jemandem verändert?
- Wann wurden diese Elemente verändert?
- Was waren die vorherigen Werte dieser Elemente?
- Welcher Typ von Element wurde verändert?

Die Protokollierung kann feingranular konfiguriert werden, um spezielle Veränderungen am Projekt aufzuzeigen. Diese Konfiguration beschränkt die Anzeige auf einen bestimmten Bereich oder Level des Modells und für eine bestimmte Zeitperiode und einen bestimmbaren Benutzer.

Elementliste

Die Elementliste ist eine editierbare Tabelle der Modellelemente, die im Hauptfenster von Enterprise Architect angezeigt werden kann. Diese Liste hilft bei der Erstellung und der Aktualisierung von Elementen eines Paketes oder Diagramms. Diese Darstellung kann zum Beispiel von Analytikern verwendet werden, um direkt in Enterprise Architect Listen von Requirements zu editieren und zu verwalten. Diese Listen können auch ausgedruckt oder als RTF-Report mit frei gestaltbarem Satz ausgegeben werden.

Nachvollziehbarkeit der Elementverwendung

An der Bedienoberfläche des Enterprise Architects kann sowohl aus dem Browser abgefragt werden, in welchen Diagrammen ein Element verwendet wurde als auch umgekehrt – in Diagrammen kann nach dem „wahren“ Speicherort eines Elements im Browser gefragt werden. Eine eigene, vorbereitete Suche listet Findelkinder (*orphans*) auf – Elemente, die in kein Diagramm eingezeichnet wurden. Dies sind zumeist Elemente, deren Löschung irrtümlich unterblieben ist.

Die hierarchische (Abhängigkeits-)Ansicht

Das “Hierarchy/Traceability”-Fenster zeigt die Abhängigkeiten des gegenwärtig selektierten Elements zu anderen Elementen. Diese Information wird von Tochterbeziehungen und/oder anderen Beziehungen zu anderen Elementen abgeleitet. Die im “Traceability”-Fenster angezeigten Beziehungen umfassen auch Aggregation, Vererbung und Abhängigkeit und Realisierungsbeziehungen. Filtertasten erlauben die Eingrenzung auf gewünschte Beziehungsarten. Dies gibt ein umfassendes Bild auf die Vernetzung, weil alle Beziehungen dargestellt werden, auch jene, die aus dem gegenwärtig betrachteten Diagramm hinausführen und beliebige Zielelemente im ganzen Modell betreffen können.

Die Matrixansicht

Die Beziehungsmatrix erlaubt die Einsicht in das Beziehungsgeflecht in Tabellenform. Darüber hinaus können in der Matrix Beziehungen auch direkt eingetragen, modifiziert oder gelöscht werden – mit einem einfachen Mausklick.

Die Abweichungsanalysematrix

Die in Enterprise Architect eingebaute Abweichungsanalysematrix hilft dabei die Lücken zwischen dem Istzustand und dem Sollzustand von Architekturen zu modellieren. Die Matrix bietet ein flexibles Werkzeug zur Definition und Überwachung identifizierter abweichender Elemente.

Diagrammerstellung mittels Elementeverkettung

Enterprise Architect kann Diagramme automatisch erweitern, ausgehend von einem gewählten Element – durch Vorgabe der Verkettungsart und der Verkettungsrichtung können „logisch benachbarte“ Elemente selektiv und automatisch in das Diagramm eingefügt werden. Diese ‘*Insert Related Element*’-Funktion ist ein mächtiges Werkzeug zur Erzeugung von Übersichtsdiagrammen aus reverse engineerem Quellcode oder reverse ermittelten Datenbankstrukturen.

Modellsuche

Mit der flexiblen Suchfunktion des Enterprise Architects können Sie nach beliebigen Zeichenfolgen oder Worten im gesamten Projekt suchen. Der zusätzliche, ausführliche Suchmodus erlaubt die Vorgabe spezieller Suchkriterien, die auch Indizierungen nutzen und rasch zu Ergebnisanzeigen führen. Wahlweise können Sie auch SQL-Abfragen dialoggestützt eingeben. Suchergebnisse können mit dem Rich-Text-Reportgenerator gedruckt oder in eine Datei geschrieben werden.

Modellsichten

Auszüge des Projektbrowserinhalts können durch filternde Anwendung frei definierbarer Regeln in einem eigenen Fenster dargestellt werden. Dies erlaubt z. B. eine Gruppierung von Elementen nach speziellen Merkmalen. Modellsichten können lokal am Anwenderrechner gespeichert werden, um „private“ Sichten auf ein Modell zu ermöglichen. Diese „privaten“ Sichten können auch exportiert und importiert werden, um anderen Nutzern zur Verfügung zu stehen oder sie werden im Speicher für gemeinsame Sichten abgelegt.

Andere Berichte

Enterprise Architect enthält mehrere, vorformulierte, nützliche Reportfunktionen:

- Ressourcen- und Planungsdetails
- Projektvorfälle
- Projektwörterbuch
- Projektgrößenstatistik

- Abhängigkeits- und Implementationsdetails
- Testdetails

Entwicklung und Zusammenarbeit im Team

Enterprise Architekt bietet Funktionalitäten, um Projekte gemeinsam und zeitparallel im Team und in verteilten Entwicklungsumgebungen abzuwickeln. Projekte können gemeinsam über Netzwerk, mit Modell-Repositories, Replikation, XMI Import/Export, Versionierungssystemen, Package-Versionierung und Sicherheits-/Lockingmechanismen betrieben werden. Das eingebaute Rechtesystem erlaubt auch Workflowberechtigungen einzurichten, sodass Review- und Freigabeprozesse auf berechnigte Personen eingeschränkt werden können – Änderungen von Status und Tagged Values mit eingeschlossen.

Ein Arbeiten über Netzwerk kann mit .EAP-Files (dateibasiert) oder mittels eines Datenbankservers erfolgen.

Unterstützung großer Modelle oder vieler gleichzeitiger Anwender

Die *Corporate* Edition und höhere Editionen von Enterprise Architekt erlaubt auch das Speichern und Bearbeiten von Modellen mittels eines DBMSs, anstelle des standardmäßigen .EAP-Files. Enterprise Architect unterstützt folgende Datenbanksysteme (DBMS):

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

XML Import/Export

Die XMI Import/Export Funktion kann benutzt werden, um einzelne Modellpackages an verschiedene Entwickler zu verteilen und sie untereinander auszutauschen. XMI erlaubt den Export von Packages als XML-Dateien, die anschließend in ein anderes EA-Projekt importiert werden können.

Die in Enterprise Architekt vorhandene Integration von Versionierungssystemen basiert ebenfalls auf XMI Dokumenten. Ein EA-Package kann unter Versionskontrolle gestellt werden, dabei wird für den Benutzer transparent ein XMI-Export durchgeführt und diese Datei in ein Versionierungssystem eines Drittanbieters eingespeichert.

Sicherheit/Locking

Die Zugriffssteuerung von Enterprise Architekt kann benutzt werden, um den Zugang zu einzelnen Funktionen im Modell für einzelne Anwender oder Anwendergruppen zu begrenzen. Mittels Locking kann die gleichzeitige Bearbeitung ein und desselben Elements oder Diagramms durch mehrere Nutzer verhindert werden. Elemente können pro Anwender oder pro Anwendergruppe gesperrt werden; diese abgestufte Sperrmöglichkeit wird durch die Anmeldung am Repository mittels - zu vergebender – Benutzer-/Passwortkombinationen sichergestellt.

Die Locking-/Sicherheits-Funktion ist nur ab der Enterprise Architect Corporate Edition verfügbar. Jeweils gesperrte Elemente werden im Browser gekennzeichnet – farblich unterschieden nach eigenen und fremden Sperren. Die eigenen Sperren können als Liste angezeigt werden. Im

Kollisionsfall - ein Anwender möchte eine Sperre auf ein Element setzen, das einer Fremdsperre unterliegt – wird angezeigt, wer der Inhaber der kollidierenden Sperre ist.

Das Modell-Diskussionsforum

Das Modelldiskussionsforum erlaubt den Anwendern, den Entwicklungsprozess und den Fortschritt eines Projektes zu diskutieren. Teammitglieder können Nachrichten lesen und Nachrichten einfügen – und können Nachrichten direkt mit Modellelementen verlinken. Bei Teams mit verteilter Entwicklung können die Anwender das EA-Modell mit einem externen Diskussionsforum auf einem Remote-Server verbinden.

Unterstützung von Frameworks für die Unternehmensarchitektur

Sparx Systems unterstützt standardisierte Frameworks, um Unternehmensmodelle zu erstellen. Frameworkimplementierungen werden in Enterprise Architect mit Hilfe von UML und seinen Erweiterungsmöglichkeiten realisiert. Dadurch wird die Einhaltung eines Standards sichergestellt. Die erstellten Modelle können mithilfe der XMI Export/Import-Funktionalität ausgetauscht werden. Die folgenden Architekturframeworks werden von Enterprise Architect als Plug-in unterstützt:

- Das Zachman Framework (siehe http://www.sparxsystems.at/MDG_Zachmann)
- DoDAF (siehe http://www.sparxsystems.at/MDG_Dodaf_Modaf)
- MODAF (siehe http://www.sparxsystems.at/MDG_Dodaf_Modaf)
- The Open Group's TOGAF (siehe http://www.sparxsystems.at/MDG_Togaf)

Alle unterstützen das Federal Enterprise Architecture Framework (FEAF) Referenzmodell

Unterstützung von SOA (Service Oriented Architecture)

Enterprise Architect erlaubt das rasche Modellieren, das *Forward Engineering* und *Reverse Engineering* zweier W3C XML Schlüssel Technologien: XML Schema (XSD) und Web Service Definition Language (WSDL).

Die Unterstützung von XSD und WSDL sind wichtig für die Entwicklung einer vollständigen Service Oriented Architecture (SOA). Die Verbindung von UML 2.1 und XML stellt den naheliegenden Mechanismus zum Spezifizieren, zum Aufbau und zur Verteilung von XML-basierten SOA-Ausprägungen in einer Organisation dar.

XSD Funktionalität

XML Schemas werden auf Basis des UML-Klassendiagramms unter Verwendung der *XML-Schema-Toolbox* modelliert. Die *XML-Schema-Toolbox* bietet das UML-Profil für XSD. Dieses gestattet die automatische Erzeugung einer W3C XML Schema(XSD)-Datei aus einem abstrakten UML-Klassenmodell.

WSDL Funktionalität

Enterprise Architect unterstützt Forward- und Reverse Engineering der W3C Web Service Definition Language (WSDL). Die WSDL-Toolbox des EAs kann zum einfachen Modellieren von WSDL-Dokumenten verwendet werden, die als Komponenten mit dem Stereotyp *WSDL* gekennzeichnet werden. WSDL-Dokumente werden in einer Package-Hierarchie abgelegt, die den WSDL Namespace und die zusammensetzenden XSD-Typen, Meldungen, Port Typen, Binding und Dienste repräsentieren.

Codeumsetzung und Codedarstellung

Die Codeumsetzung umfasst automatische Codeerstellung, Code-Rückumsetzung (*Reverse*

Engineering) aus Quellcode und die Synchronisation zwischen Quellcode und Modelldarstellung. Dies ist allerdings nur in der Professional- und in der Corporate Edition und höheren Editionen des Enterprise Architects enthalten. Besitzer der Desktop-Edition können diese auf der Upgrade-Seite http://www.sparxsystems.com/registered/reg_ea_upgrade.html aufrüsten.

Der Enterprise Architect ermöglicht Ihnen die **Quellcodeerstellung unmittelbar aus UML-Modellen** in mehr als zehn gebräuchlichen Programmiersprachen, einschließlich:

- ActionScript (Macromedia Flash Entwicklungssprache)
- Ada
- C
- C# (sowohl .NET 1.1 als auch .NET 2.0)
- C++ (Standard und .NET bedingte C++ Erweiterungen)
- Delphi
- Java (einschließlich Java 1.5, Signalisierung und Gattungen)
- PHP
- Python
- System C
- Verilog
- VHDL
- Visual Basic
- Visual Basic .NET

Die Quellcodeerstellung schließt Klassendefinitionen, Variablen und Funktionsrümpfe für jedes Attribut und jede Methode einer UML-Klasse mit ein. Zusätzlich erlaubt der EA XSD und WSDL XML-Quellcode zu erzeugen und zu editieren, (siehe auch Abschnitt *SOA* in diesem Dokument).

Die Codeerstellung wird auch als **Forward Engineering bezeichnet**. Der Import von Quellcode in eine Modelldarstellung wird auch **Reverse Engineering** genannt. Alle für Quellcodeerzeugung eingebauten Sprachen sind auch für Reverse Engineering im EA verfügbar.

Von **Synchronisation** spricht man, wenn Änderungen im Modell in den Quellcode exportiert werden oder wenn Änderungen im Quellcode in das Modell übernommen werden. Dies ermöglicht Ihnen, Ihr Modell und Ihren Quellcode abgeglichen auf dem letzten Stand zu halten, wenn sich das Projekt weiterentwickelt.

Als **Round Trip Engineering** bezeichnet man die Kombination von Reverse und *Forward Engineering* von Code als auch die Synchronisation, ein typischer Vorgang in Softwareprojekten.

Die EA-Editionen „Business & Software Engineering“, „Systems Engineering“ und „Ultimate“ können auch aus allen **UML-Verhaltensdiagrammen forward engineer**, d. h. *innere, ausführende* Codestrukturen erzeugen. Die Forward-Übersetzungsregeln sind durch erweiterte Codetemplates zugänglich und veränderbar. Das mitgelieferte EAExample.eap enthält ausführliche Beispiele hierzu. Diese erweiterte Forward-Generierung ist auch für C verfügbar.

Import von .jar files und .NET assemblies

Enterprise Architect (Professional und Corporate Edition) gestatten das Reverse-Engineering folgender binärer Module:

- Java Archive (.jar)

- .Net PE Dateien (.exe, .dll); native Windows DLL und Exe Dateien werden nicht unterstützt, nur PE Dateien, die .Net Assembly Daten enthalten.
- Intermediate Language Dateien (.il).

Vorlagengetriebene Codeerstellung

Sie benutzen grundsätzlich das **Code Template Framework** (CTF) - die EA-Vorlagenverwaltung - beim Forward-Engineering eines UML-Modells. Diese Templates bestimmen dabei eine adaptierbare Umsetzung von UML-Elementen in verschiedene Programmiersprachen.

Das **Code Template Framework** ermöglicht Ihnen:

- Quellcodeerstellung aus UML Modellen.
- Anpassung der Art und Weise wie EA den Quellcode erstellt.
- Das Forward-Engineering von Sprachen, die nicht im originalen Lieferumfang des EA enthalten sind.

Das Code Template Framework besteht aus:

- Standard-Codevorlagen, die bereits in EA für die unterstützten Sprachen eingebaut sind
- Einem Code-Editor zum Erstellen und Warten benutzerdefinierter Codeumsetzungsvorlagen.

Angestoßene und automatische “live”-Codeerstellung

Enterprise Architect stellt eine “Live Code Generation” Funktion zur Verfügung, die automatisch Ihren Quellcode auf den letzten Stand bringt, wenn Sie Änderungen in ihrem Modell vornehmen. Zum Beispiel, wenn Sie neue Methoden oder Attribute in ihrem Klassenmodell anlegen – diese werden dann automatisch in ihren Quellcode hinzufügend eingetragen.

Eingebauter Codeeditor mit Hervorhebungen und dynamischer Strukturanzeige

Sie können den eingebauten Codeeditor zur Anzeige und zum Ändern von Quellcode verwenden. Wenn Sie ein Element mit einer zugewiesenen Quellcodedatei im Modell auswählen, wird der Quellcode in einem Editor mit Grammatik-Hervorhebung und einer navigierbaren Code-strukturanzeige angezeigt. Die Quellcodeanzeige bietet auch eine Werkzeugleiste für schnelle Codeerstellung und Synchronisation mit dem Modell.

Debugging, Compiling und Darstellung von Programmen

Die Professional und die Corporate Edition und die höheren Editionen des Enterprise Architects erlauben das Erstellen, Testen, Debuggen und Ausführen von Code innerhalb des Enterprise Architects.

Diese Funktion führt für den Entwickler UML-Modell- und Codeerstellung noch enger zusammen. Mit der Möglichkeit, NUnit- und JUnit-Testklassen aus Implementierungsklassenmodellen mittels MDA-Transformation zu erzeugen und der Möglichkeit, den Testprozess direkt im EA-durchzuführen, ist es für den Entwickler noch leichter, mit geringem Aufwand die Dokumentation immer up-to-date zu halten.

Zusätzlich zu den Funktionalitäten *Erstellung* und *Test* bietet der EA *Debugging-Möglichkeiten* für .NET und Java. Die Fähigkeit des Enterprise Architects, Debugger im Kommandomodus anzusprechen, wurde entwickelt, um für Entwickler und Tester Stackinhalte beim Durchlauf durch den ausführbaren Code zu sammeln. Das gesammelte Stacklog wird dann im EA benutzt, um Sequenzdiagramme zu erstellen, d. h. die aktuell aufgetretenen Programmaufrufe in Diagrammform zu visualisieren. Dies ist eine äußerst zeitsparende und effiziente Technik mit komplexen Projekten umzugehen, weil hier verlässlich und leicht lesbar dokumentiert werden kann, dass sich der erstellte

Code genau so verhält, wie vom Architekten und Entwickler geplant.

Erstellen und Ausführen von Scripts

Jedes Package in einem EA-Projekt kann konfiguriert werden, wie es kompiliert und ausgeführt werden soll. Dabei ermöglicht EA das Speichern mehrerer Konfigurationen pro Package. Enterprise Architect bietet auch die Möglichkeit, die Ausgaben des Compilers zu parsen und das Codeeditorfenster zu öffnen, um diese Fehler zu finden.

Kompilieren und Ausführen mit der Debug Workbench

Enterprise Architect erlaubt es, pro Package eine oder mehrere Kompilierungs- und Ausführungskonfigurationen (*build* and *run*) zu definieren. Weiters können Scripts erstellt werden, um einen Testprozess oder Verteilungsprozess (*deployment*), sowie ein Debugging des Codes anzustoßen. Zur Visualisierung des Sourcecodes wird der im EA integrierte Codeeditor verwendet.

Automatische Erzeugung von Sequenzdiagrammen

Es ist einfach, detaillierte und auch umfangreiche Sequenzdiagramme automatisch aus dem aufgezeichneten Debuggingdurchgang zu erzeugen.

Voraussetzung ist ein Modell und der dazu passende, ausführbare Code. Das Modell befindet sich in einem Package, für welches die Debuggingfunktionalität im EA konfiguriert wurde. Der Code, der durch ein in diesem Package enthaltenes Klassendiagramm erstellt wurde, kann „innerhalb“ des EA ausgeführt werden, wodurch ein *stack-trace* erstellt wird, aus dem anschließend automatisch ein Sequenzdiagramm generiert werden kann. Damit haben Sie die Möglichkeit, Ihr lauffähiges System zu dokumentieren und dessen Arbeitsweise zu überprüfen.

EA erlaubt Ihnen dabei entweder manuell durch das Programm durchzusteppen oder den Debugger automatisch durchlaufen zu lassen.

Eingebaute Unterstützung für JUnit und NUnit Tests

Enterprise Architect erlaubt es Testspezifikationen zu erstellen, Tests durchzuführen und die Ergebnisse aufzuzeichnen. Die gemeinsame Verwaltung von Modellelementen und Testfällen kann die Kommunikation zwischen Softwareentwicklung, Analyse, Design und Qualitätsmanagement verbessern.

Der erzeugte Code kann anschließend mit der Kompilier- und Ausführungs- (*build* and *run*) Funktionalität von Enterprise Architect erstellt werden. Alternativ können Sie auch ein Testscript definieren, welches die Testfälle ausführt.

Enterprise Architect bietet dabei Tools für manigfaltige testbezogene Abläufe:

- Testpoint-Verwaltung, die eine mächtige modelbasierte Testdurchführung bietet. Testpoints können auf Elementen des Systemdesigns definiert werden und auch durchgeführt werden. Die Ergebnisse werden in Echtzeit, während die Anwendung läuft.
- Testspezifikation – für jedes Modellelement können in Enterprise Architect Testspezifikationen definiert werden. Ebenso wie die Informationen, wer den Testfall durchgeführt hat und wann der Durchführungsteitpunkt war. Unit-, System-, Integrations- sowie Akzeptanz und Szenarien-Tests können erstellt und überwacht werden. Dadurch entsteht eine enge Verknüpfung zwischen Systemarchitektur und Testinformation.
- JUnit- und NUnit-Unterstützung: EA unterstützt Sie dabei, diese Testklassen mit Hilfe automatischer Transformationen zu erstellen. Das Ergebnis dieser Transformation ist ein Gerüst von Testklassen für Ihr Implementierungsmodell. Lediglich die Testlogik muss noch manuell implementiert werden.

Integration des Unit-Testings in den Entwicklungszyklus

Ein Grundprinzip der Unit-Tests besteht darin, die Tests vorab zu schreiben. Enterprise Architect hilft, dieses Prinzip einzuhalten. Beim Hinzufügen einer neuen Methode zu einer Klasse führt man die Transformation der Klasse aus. EA erzeugt die korrespondierende Methode und Sie können die Details festlegen und einfügen. Dies kann einfach erfolgen, noch bevor der zu testende Code selbst erstellt wird.

Performanz und Skalierbarkeit

Enterprise Architect ist schnell! EA Benutzer stimmen darin überein: Der EA lädt Projekte in einem Bruchteil der Zeit vieler anderer am Markt befindlichen Werkzeuge und er verarbeitet auch extrem große Modelle ohne nennenswert an Performanz einzubüßen.

Versionierung

Enterprise Architect unterstützt auch die Versionierung von Packages und darin enthaltener Sub-Packages in einem zentralen VCS-Repository. Dieses Repository wird von Drittanbietern bereitgestellt und ist selbst nicht Bestandteil von EA. VCS' werden aber von EA für den Benutzer transparent integriert.

Die Versionierungssteuerung des EA bietet zwei Schlüsselmerkmale:

- Unterstützung der gemeinsamen Nutzung von Packages zwischen einzelnen Benutzern.
- Speicherung des Entwicklungsverlaufs, die einzelnen Versionen werden gespeichert, auch ältere können abgerufen werden.

Enterprise Architect unterstützt folgende Versionierungssysteme:

- Alle Versionierungssysteme kompatibel mit dem "Microsoft Common Source Code Control Standard", ab der Version 1.1. (Zum Beispiel Visual Source Safe oder Clear Case)
- Microsoft Team Foundation Server (TFS)
- Subversion, verfügbar unter <http://subversion.tigris.org>
- CVS, verfügbar <http://www.wincvs.org>

Baselines und Versionsvergleich

Enterprise Architect bietet ab der Corporate Edition eine Basislinie- (*Baseline / snap-shot*) Funktionalität an, die den Modellzustand zu einem gewünschten Zeitpunkt festhält. Baselines werden im XMI-Format abgelegt und gemeinsam mit dem Modell in komprimierter Form im EA-Repository gespeichert. Es können auch mehrere Baselines für ein einzelnes EA-Package gespeichert werden. Eine zu einem bestimmten Projektzeitpunkt abgespeicherte Basislinie kann später dazu verwendet werden, einen Vergleich des aktuellen Packagezustandes mit der Baseline anzuzeigen.

Die Differenzermittlungsfunktion der Professional, Corporate und der höheren Editionen zeigt auf, was zwischenzeitlich geändert wurde und wie sich dies von der Version zum Zeitpunkt der Basislinie unterscheidet. Diese Funktion erlaubt Ihnen einen Modellzweig im EA zu vergleichen, mit:

- einer Basislinie, die mit der Baselinefunktionalität erstellt wurde.
- einer Datei auf einem Datenträger, die zuvor mit der EA-XMI-Exportfunktion erzeugt wurde.
- einer XMI-Datei aus dem (externen) Versionierungssystem, die aus dem EA heraus erstellt worden ist.

- Einer Baseline in einem Package eines anderen EA-Repositories, von dem das aktuelle Package initial als Kopie in ihrem Projekt erzeugt wurde. Letztere Variante wird vor allem bei Branching eingesetzt.

Zusammenspiel und Integration mit anderen Werkzeugen

Enterprise Architect bietet mehrere Mechanismen, um Ihre Modelle in Werkzeuge anderer Hersteller überzuführen, einschließlich einer programmierbaren API, sowie ein in EA integriertes Transformationsframework. Durch das integrierte Transformationsframework werden bereits vordefinierte Transformationen mitgeliefert.

Automation Interface

Das "Automation Interface" erlaubt Ihnen den Zugriff auf die EA-internen Modelle aus selbst geschriebenen Programmen. Als Beispiel seien einige Aufgaben genannt, die durch dieses Interface gelöst werden können:

- Ausführung iterativer/rekursiver Aufgaben, wie z. B. die Veränderung der Versionsnummer aller in einem Modell enthaltenen Elemente.
- Automatische Codeerzeugung aus einem State-Machine-Diagramm nach eigenen, speziellen Wünschen.
- Erzeugung benutzerdefinierter Spezialreports.
- Ad-Hoc-Abfragen.

Alle Entwicklungsumgebungen, die ActiveX COM Clients erzeugen, können mit dem Enterprise Architect Automation Interface arbeiten.

Add-ins

Add-ins erlauben Ihnen, weitere Funktionalität in Enterprise Architect hinzuzufügen. Das Enterprise Architect Add-in-Framework baut auf den Funktionen des Automation Interfaces auf und erlaubt, das Benutzer-Interface des EA zu erweitern. Add-ins bieten einige wesentliche Vorteile gegenüber abgesetzten Erweiterungslösungen - einschließlich der Möglichkeit, erweiterte EA-Menüs zu definieren und der Möglichkeit der Übernahme von Ereignissen aus der EA-Bedienungsumgebung, wie Klicks in Menüs und von Auswahlen durch den Benutzer.

MDG Link

Sparx hat eine Anzahl an *Model Driven Generation* (MDG)-Produkten entwickelt, um ein direktes Zusammenspiel mit anderen Werkzeugen zu ermöglichen. MDG Link Produkte zeigen beispielhaft den Gebrauch des eingebauten „Add-in Frameworks“, um die Funktionalität des Enterprise Architects wesentlich zu erweitern. Der MDG Link für Visual Studio und der MDG Link für Eclipse ermöglichen dem Enterprise Architect mit dem Microsoft® Visual Studio® und Eclipse IDE's direkt zu interagieren.

MDG Integration

MDG Integration für Visual Studio 2005/ 2008/2010 vereint Enterprise Architect mit der Microsoft® Visual Studio® 2005/ 2008/2010 Entwicklungsumgebung. Dieses Produkt ermöglicht Anwendern, das UML-Modell innerhalb von Visual Studio einzusehen, zu editieren und stellt auch viele der Hauptfunktionen des Enterprise Architects direkt in der Entwicklungsumgebung zur Verfügung, einschließlich der Rich-Text- und Web-Dokumentenerstellung, der MDA Transformationen, des *Baseline* Managements und der XML-basierten Techniken.

Auch für Eclipse-basierte Umgebungen steht eine MDG Integration zur Verfügung. Dabei werden unter anderem Adobe® Flex® Builder™ und Progress OpenEdge® Architect unterstützt.

MDA (Model Driven Architecture) Unterstützung

Enterprise Architect bietet die Möglichkeit zur Ausführung von MDA-Transformationen. Er unterstützt eine voll konfigurierbare Methode, Modellelemente oder Modellfragmente von einer Domäne in eine andere zu konvertieren. Das betrifft typischerweise die Umwandlung eines plattformunabhängigen Modells (PIM) in ein plattformspezifisches (PSM) Modell. Ein einzelnes Element eines PIM kann so der Ausgangspunkt für mehrere Elemente in einzelnen PSMs sein, sowie in verschiedene Domänen übersetzt werden.

Transformationen sind ein deutlicher Produktivitätsgewinn und reduzieren den Aufwand, manuell Stammklassen und Elemente einer einzelnen Implementationsdomäne aufzusetzen; zum Beispiel können Datenbanktabellen automatisch aus PIM Klassen abgeleitet werden.

Die MDA Funktionalität von Enterprise Architect erlaubt folgende Aktionen:

- Das automatische Generieren von folgenden Modellen:
 - Datenmodellierung (DDL)
 - Code Modelle, wie z. B. Java, C#
 - XML Modelle, wie z. B. XSD und WSDL
 - Testmodelle für JUnit und NUnit Tests
 - EJB Entity
 - EJB Session
- Das Erstellen von benutzerdefinierten Transformationen, basierend auf einem mächtigen Templatemechanismus.
- Wiederholtes Ausführen von Transformationen, um ein geändertes Source-Modell in ein neues Target-Modell überzuführen.

Datenbankmodellierung

Verfügbares Datenbank-Modellierungsprofil

Datenbankmodellierung und Datenbankdesign werden nicht ausdrücklich durch die UML-Spezifikation abgedeckt. Enterprise Architect stellt allerdings eine Erweiterung der UML zur Verfügung (UML-Datenmodellierungsprofil) um Relationenmodelle erstellen zu können. Dieses Profil bietet einfach verwendbare und leicht verständliche Erweiterungen des UML-Konzepts, einer Abbildung von Datenbankstrukturen mit ihren Tabellen und Beziehungen auf Klassen und Assoziationen. Diese Erweiterung erlaubt Ihnen auch, Datenbankschlüssel, Trigger, Begrenzungen, RI und andere Eigenschaften relationaler Datenbanken abzubilden.

Beim Datenbankentwurf werden folgende typischen Vorgänge auftreten:

- Erstellen eines Datenmodell-Diagramms
- Anlegen einer Tabelle
- Eigenschaften einer Tabelle festlegen
- Spalten anlegen
- Primäre Schlüssel anlegen
- Fremdschlüssel anlegen
- Anlegen von Stored Procedures

- Anlegen von Indizierungen und von Triggern
- Erstellen einer DLL für eine Tabelle
- Erstellen der DLL für ein ganzes Package
- Datentypumwandlungen in einer Tabelle
- Datentypumwandlungen für ein ganzes Package
- Anpassen der Datentypen für eine Datenbank
- Importieren eines Datenbankschemas aus einer ODBC-Datenquelle
- Anlegen von Views (SQL Server, Oracle und Adaptive Server Enterprise).

Durch die in EA vorhandenen MDA Transformationen ist es möglich, aus einem Klassendiagramm automatisch ein Relationenmodell erstellen zu lassen. Dabei wird das Klassendiagramm als das bekannte *Entity-Relationship* Modell interpretiert.

Unterstützung gängiger Datenbanksysteme

Der EA unterstützt die Modellierung mit Schemaerstellung zu folgenden Datenbanken:

- DB2
- InterBase
- Informix
- Ingers
- MS Access 2007
- MySQL
- Oracle 9i, 10g und 11g
- PostgreSQL
- SQL Server 7 and MS SQL Server 2000, 2005 and 2008
- Sybase Adaptive Server Anywhere
- Sybase Adaptive Server Enterprise
- Firebird.

DDL Erstellung (Eingangsdatei für den Schema-Prozessor)

Enterprise Architect verfügt über einen Generator, der aus einem Datenmodell DDL Code erzeugen kann. Es werden standardmäßig 9 verschiedene DBMS' unterstützt.

Import von Datenbankstrukturen aus ODBC-Verbindungen

Durch das Rückführen (*reverse engineering*) über ODBC Schnittstelle, können Datenbanksysteme analysiert werden. Dabei wird ein Relationenmodell der laufenden Datenbank erstellt. Dieses Relationenmodell können Sie immer wieder neu importieren oder eine Synchronisation mit einem vorhandenen Modell ausführen. Verlinkungen ins Modell gehen dadurch nicht verloren.

Unterstützung für Systems Engineering

Die EA Versionen Ultimate und Systems Engineering stellen viele nützliche Funktionen wie etwa die vollständige Unterstützung von SysML 1.2, Parametrische Modellsimulation, die Generierung ausführbaren Codes, die modellbasierte Transformation für Hardwarebeschreibungs-Sprachen aber auch ADA 2005.

SysML und Modellsimulation

Enterprise Architect erlaubt es SysML Modelle rasch und einfach zu erstellen. Dies wird durch ein in Enterprise Architect integriertes SysML-Profil erreicht. Dabei werden alle neun Diagrammtypen aber auch Modellvalidierungsregeln erreicht. Die Simulation parametrischer Modelle unterstützt die Analyse kritischer Systemparameter. Darunter auch Metriken wie etwa Performanz oder Zuverlässigkeit eines Systems.

Fortgeschrittene Verhaltensmodellierung und Generierung ausführbaren Codes

Enterprise Architect kann UML Aktivitäten mit Interaktionselementen verknüpfen und erlaubt es damit ausführbaren Code aus Interaktions- (Sequenz-) und Aktivitätsmodellen zu generieren. Das erlaubt es, mehr als nur Codestubs aus Verhaltensmodellen abzuleiten. Durch das Generieren aller eingebetteten Modelle, die ausführbare Anweisungen enthalten.

Hardwarebeschreibungssprachen

Die Enterprise Architect Versionen Systems Engineering und Ultimate unterstützen die Codegenerierung aus Zustandsautomaten in HDL wie etwa VHDL, Verilog oder SystemC.

Unterstützung der Projektabwicklung

Enterprise Architect bietet Projektmanagementunterstützung. Projektmanager können Enterprise Architect verwenden, um einem Element (Use Case, Klasse, etc.) Ressourcen zuzuweisen, Risiken und Aufwände einzutragen sowie die Projektgröße zu ermitteln. EA unterstützt auch die Abwicklung von Änderungen und Wartungsaufgaben.

Projektkennzahlenermittlung mittels "Use Case Metrics"

Die "Use Case Metrics" - Funktion in Enterprise Architect erlaubt aufgrund der Menge und Komplexität der einzelnen Use Cases eine Aufwandsschätzung für das zu erstellende System automatisch zu berechnen (*Use Case Point Method*). Die ermittelten Zahlen sind nur ein Schätzwert, können allerdings bei wiederholter Anwendung und anschließender Überprüfung als Vorhersagewerkzeug nützliche Dienste leisten. Üblicherweise ist die Genauigkeit der Aufwandsberechnung bei hinreichender Detaillierung der System-Use-Cases überraschend gut. Eine Gewichtungsspeicherung bei Requirements und Risiken ist enthalten, sodass auch die Function-Point-Methode nach einem Datenexport aus EA heraus angewandt werden kann.

Ressourcen

Enterprise Architect kann ressourcenbezogene Informationen und Basisinformationen über die Entwicklungsanteile im Modell speichern. Dieser Zugang erlaubt eine enge Kopplung zwischen Projektmanager und dem Entwicklungsteam. Die ressourcenbezogenen Informationen verschaffen dem Projektmanager den benötigten Überblick, um den Projektfortgang und auftretende Wechselwirkungen sichtbar zu machen. Sich anbahnende Planabweichungen, kritische Ressourcenverfügbarkeiten können mit direktem Bezug zu Modellelementen ermittelt werden und durch die leistungsfähige Such- und Filterfunktion des EAs leicht lokalisiert werden.

Testen

Zusätzlich zu den automatisch generierten JUnit und NUnit Testklassen erlaubt der EA die Zuordnung von beliebig vielen und beliebig komplexen Testfällen zu jedem einzelnen Modellelement. Die Modellelemente und die zugehörige Testdokumentation in einem einzigen, integrierten System zu halten, verbessert die Kommunikation zwischen dem Testteam, den Softwareentwicklern und den Softwarearchitekten wesentlich. Auch hier macht es die umfangreiche Suchfunktion leicht, fehlgeschlagene Testfälle, noch nicht begonnene Tests oder erfolgreich abgeschlossene Testfälle aufzufinden. Durch die Nutzung der Test- und Suchfunktionalitäten ist es einfach, durch das Projekt zu navigieren und die Problempunkte unmittelbar aufzufinden, Designschwachpunkte zu identifizieren oder andere kritische Umstände einzugrenzen. Der EA ist

nicht nur ein UML-Werkzeug, er ist auch ein komplettes Test Management-Werkzeug. Der EA ist auch geeignet, Testhandbücher zu erstellen und die Testabwicklung darin zu dokumentieren. Selbstverständlich können Sie die zugeordneten Testfälle im Testhandbuch frei reihen, um den Vorbereitungsaufwand für ähnliche Tests zu minimieren.

Change Management

Die Berücksichtigung nachgereicher Änderungswünsche ist immer im Zusammenhang mit dem Gesamtentwicklungsprozess zu sehen. Change Requests können im EA-Modell eingearbeitet, in ihrer Auswirkung analysiert und dokumentiert werden, ohne die aktuelle Modellierung zu stören. Eigene Changerequest-Analysereports können unabhängig erstellt werden.

Mittels transparenter oder impliziter Versionierung kann die Umsetzung der Changes durchgehend dokumentiert werden. Mittels API oder Scripting sind Automatismen zur Kennzeichnung der durch Changes betroffenen Wirkungsketten leicht einsetzbar.

Projektaufgaben

Das Setzen, Priorisieren, Zuteilen und Nachverfolgen von Aufgaben in einem Projekt ist ein kritischer Teil des Projektmanagements während einer Systementwicklung.

In Erweiterung der bereits vorgestellten Projektmanagement-Funktionalitäten, erlaubt EA auch die Verwaltung und Verfolgung von Projektaufgaben, die direkt Projektmitarbeitern und zugleich einzelnen Modellelementen zugeordnet werden. Vorlagemechanismen nach verschiedenen Sortierkriterien (Person, Budget, Abarbeitungsstand, Termin – auch kombiniert) sind eingebaut.

Auch Nachrichten können damit versandt und empfangen werden.

Wörterbuch

Als Zusatzfunktion verwaltet der EA ein Modellwörterbuch, eine einfache Möglichkeit, Fachbegriffe des Projektbereichs, die möglicherweise einzelnen Projektmitgliedern nicht geläufig sind, zentral zu definieren und zu kommunizieren. Die Verwendung von im Wörterbuch enthaltenen Begriffen in Elementebeschreibungstexten wird automatisch als interner Hyperlink angezeigt, die im Wörterbuch gespeicherte Erläuterung wird als Mouse-Over angeboten. Aus Texten können Begriffe direkt ins Wörterbuch aufgenommen werden.

Präsentationsfunktionen

In der Projektpraxis sind Modellinhalte zu präsentieren. Sehr oft enthalten Diagramme mehr Information, als in der Präsentation gezeigt werden soll. Die Funktion *diagram filters* erlaubt mittels Filterregeln Teilinhalte von Diagrammen zu schattieren oder auszublenden.

Import - Export Funktionalitäten

Ein Datenaustausch zwischen unterschiedlichen Modellen und zwischen unterschiedlichen Werkzeugen ist eine gängige Anforderung bei einem Entwicklungsprojekt. Enterprise Architect unterstützt die beiden gängigsten Datenaustauschformate (XMI und CSV) und erleichtert bei Bedarf die Informations- und Modellportierung in unterschiedliche Werkzeuge oder Repositories. Mit der EA API ist ein automatischer Import/Export nach eigenen Vorstellungen leicht einrichtbar. Die Verteilung von Modellen und die automatische Softwareerstellung werden dadurch wesentlich vereinfacht.

Unterschiedliche XMI Formate

Enterprise Architect unterstützt den Import und Export vieler XML-basierter Modellaustauschformate. Das bekannteste Austauschformat XMI (XML Metadata Interchange) ist ein offener Standard. Aufgrund der XML Struktur können Modelle leicht durchsucht, verarbeitet und verteilt

werden. Enterprise Architect unterstützt XMI 1.0, XMI 1.1 und XMI 2.1. Diese Unterstützung mehrerer Formate ist wesentlich, da viele Werkzeuge eine spezielle XMI-Version voraussetzen. Weiters können mit Enterprise Architect Modelle im Format EMX und UML2 importiert werden.

CSV

Zusätzlich zum XMI Import/Export, bietet der EA eine einfache CSV-Import/Export Funktion. Das ist hilfreich, um Inhalte an Werkzeuge wie Microsoft Excel zu übergeben oder Listen aus Tabellenkalkulationen zu übernehmen.

UML-Erweiterungen in Enterprise Architect

UML Profile

Mit UML Profilen kann die Sprache UML erweitert werden. Dieser Mechanismus gibt Ihnen die Möglichkeit, erweiterte UML-Modelle für spezielle Anwendungszwecke zu erstellen. Profile bauen auf zusätzlichen Stereotypen und Tagged Values auf, die auf Elemente, Attribute, Methoden, Beziehungen oder Beziehungsendpunkte angewandt werden. Ein Profil ist eine Sammlung solcher Erweiterungen, die zusammen eine spezielle Modellierungsaufgabe beschreiben und die Modellkonstruktion in diesem Bereich unterstützen. Als Beispiel sei das von David Carlson in *Modeling XML Applications with XML* (p. 310) entwickelte UML-Profil für XSD Schemata genannt. Mit diesem UML-Profil können mit UML in Klassendiagrammen XSD Schemata modelliert werden.

Enterprise Architect hat einen generischen UML-Profil-Mechanismus zum Laden und Arbeiten mit unterschiedlichen Profilen. UML Profile für Enterprise Architect werden mittels XML-Dateien in einem spezifischen Format definiert. Derartige XML-Dateien können in das *Resources*-Fenster des EAs eingefügt werden. Sobald das Profil in EA geladen wurde, wird automatisch eine neue Toolbox für das Profil erstellt. Der EA fügt die Stereotypen, Tagged Values und die Standardwerte, Notizen und, falls vorhanden, auch eine Metadatei automatisch in das neue erstellte Element ein. Sie können auch Attribute und Operationen in bestehende Klassen ziehen und dadurch direkt hinzufügen – mit dem vorgegebenen Stereotyp, den Werten und anderen Parametern. Auch die gewünschten Icons für die Toolbox können selbst definiert werden. *Shape-Scripts* können zur Anzeigestaltung der neuen Elemente eingesetzt werden.

UML Muster (Patterns)

UML Muster/Patterns sind parametrisierte „best practice“ Lösungen für immer wieder kehrende Probleme; d. h. es handelt sich um eine Gruppierung von zusammengefassten Objekten und Klassen. Enterprise Architect stellt die geläufigsten UML Muster als Templates zum Einfügen in Ihre Projekte zur Verfügung. Durch die Anwendung von Mustern erhalten Sie ein robustes und leicht wartbares System. EA bietet auch die Möglichkeit, eigene Muster als Vorlage zu speichern. Sobald die Verwendbarkeit eines Musters in einem neuen Projekt erkannt wird, kann das eingefügte Muster aus früheren Projekten mit angepassten Variablennamen im gegenwärtigen Projekt verwendet werden.

Patterns beschreiben generisch die Lösung eines abstrakten Problems und es ist die Aufgabe des Anwenders eines Patterns, dieses an den konkreten Bedarf im aktuellen Projekt anzupassen.

MDG Technologie

Die *Model Driven Generator (MDG) Technologie* erlaubt Ihnen die Bündelung einer logischen Vorlagensammlung einer spezifischen Technologie an zentraler Stelle im Enterprise Architect. Mit der MDG Technologie bekommen Sie die Möglichkeit, UML-Profile, UML-Patterns, Codevorlagen und Codeteile gut gegliedert in der *Resource View* des Enterprise Architects abzulegen. Alle notwendigen Teile zur Verwendung der MDG Technologie finden Sie als Angebot von Sparx

Systems zum Download unter from: www.sparxsystems.com/resources/mdg_tech.

Spezielle Add-Ins

Add-Ins fügen Enterprise Architect weitere Funktionen hinzu. Die Enterprise Architect Add-In Funktionalität stützt sich auf das erprobte Automationsinterface, sodass Sie auch das Benutzer-Interface erweitern können.

Add-Ins sind ActiveX COM Objekte, die öffentliche Aufrufmethoden zur Verfügung stellen. Dies hat verschiedene Vorteile gegenüber abgesetzten Automatisierungslösungen:

- Add-Ins können EA-Menüpunkte und Untermenüs anlegen.
- Add-Ins erhalten Bestätigungen über verschiedenste EA-Benutzer-Interface Vorgänge, einschließlich Klicks in Menüs und Dateiwechsellvorgänge.
- Add-Ins können (und sollten) als Prozesskomponenten (DLL) geschrieben werden; dies bedingt weniger Aufrufaufwand und bessere Integration in die EA-Umgebung.
- Läuft schon eine Instanz des EAs, gibt es keinen Grund, eine zweite Kopie über das Automationsinterface zusätzlich zu starten.
- Da das Add-In der aktuell laufenden Kopie des EAs zugeordnete Object Handles erhält, ist mehr Information über das aktuelle Benutzerverhalten verfügbar – z. B. über gerade im Diagramm selektierte Elemente.
- Um Add-Ins verfügbar zu machen, genügt es, sie zu installieren; es ist keine Konfiguration erforderlich.

Enterprise Architekt Editionen

Enterprise Architect ist in verschiedenen Editionen verfügbar; Ultimate, Systems Engineering, Business & Software Engineering, Corporate, Professional und Desktop. Für die ersten vier Edition können wahlweise persönliche oder Floating-Lizenzen erworben werden. Jede dieser Editionen stellt Funktionen zur Verfügung die das alleinige Arbeiten bis hin zum Arbeiten in großen Teams unterstützt. Die Floating Lizenzen sind speziell für Firmen nützlich, welche einen Pool an Lizenzen verwalten wollen, um EA Lizenzen von mehreren Benutzern zu unterschiedlichen Zeiten benutzen zu können.

Die Verwaltung der Floating Lizenzen kann auch als Dienst unter Windows auf einem Server eingerichtet werden. Im Log dieses Dienstes kann nachvollzogen werden, wie oft und wie viele Lizenzen zugleich benutzt wurden.

Weitere Informationen über Enterprise Architekt Editionen finden Sie unter:

<http://www.sparxsystems.eu> Prozessunterstützung

UML ist eine Sprache, kein Prozess. UML beschreibt die Elemente einer Modellierungssprache und wie diese Elemente zusammengesetzt werden können, um Umstände in einer realen Welt zu repräsentieren. UML beschreibt nicht, wie diese Elemente der Reihe nach einzusetzen sind, um ein neues Softwaresystem aufzubauen.

Wie UML ist der EA *prozess-neutral*, d. h. er enthält alle Bestandteile und Funktionen um einen gewünschten Entwicklungsprozess umzusetzen, aber er schreibt nicht vor, wie dieser Prozess aussehen soll oder umzusetzen ist.

Viele EA-Anwender wenden hochstrukturierte Prozesse an, z. B. den Rational Unified Process (RUP), während andere Anwender flexiblere, agilere und weniger restriktive Prozesse bevorzugen. Unabhängig davon, welchen Grad der Prozessorientierung Sie wünschen, der EA enthält die benötigten Werkzeuge und Bestandteile um den fortschreitenden Prozess einer Softwareentwicklung zu handhaben.

Über Sparx Systems



Sparx Systems wurde 1996 von Geoffrey Sparks in Creswick Australien gegründet. Sparx Systems ist beitragendes Mitglied der *Object Management Group* (OMG), dem verantwortlichen Standardisierungsgremium der UML-Spezifikation und vieler anderer, zugehöriger Spezifikationen.

SparxSystems Central Europe Software GmbH ist seit Mai 2004 in Europa aktiv, wobei zunächst mit der Übersetzung für den deutschsprachigen Markt begonnen wurde. Nunmehr haben wir uns als lokaler Ansprechpartner für den gesamten deutschsprachigen Raum etabliert.

Firmen-Vision

Sparx Systems ist der Überzeugung, dass ein umfassendes Modellierungs- und Entwurfswerkzeug für den *gesamten* Softwarelebenszyklus eingesetzt werden soll. Unsere langfristige Planung spiegelt dies wieder, ebenso wie unsere Überzeugung, dass eine Software für den Lebenszyklus genau so dynamisch und modern wie die von Ihnen entworfenen und betriebenen Systeme sein soll.

Die Software von Sparx ist zum Einsatz durch Analysten, Designer, Architekten, Entwickler, Tester, Projektmanager und die Wartungsmannschaft gedacht; das ist also praktisch jeder, der in einer Softwareentwicklung oder einer Analyse beteiligt ist. Es ist die Überzeugung von Sparx, dass hochpreisige CASE-Tools in ihrer Sinnhaftigkeit für ein Team und damit für eine Organisation stark beschränkt sind, weil sie durch ihre Kosten den einfachen Zugang zum Modell und damit zum Entwicklungswerkzeug behindern. Daher fühlt sich Sparx Systems verpflichtet, einerseits eine akzeptable Preispolitik zu betreiben und andererseits einen EA-Reader kostenlos für jene zur Verfügung zu stellen, die nur die Einsicht in ein Modell benötigen.

Langfristige Ausrichtung auf unternehmensweite UML Werkzeuge

Sparx Systems hat über zehn Jahre Modellierungswerkzeuge entwickelt und hat dabei den Enterprise Architect auf unternehmensweite Verwendung ausgelegt. Zusätzlich bemüht sich Sparx Systems, Anforderungen und Vorschläge aus der breiten Anwenderschar zu berücksichtigen. Dadurch konnten schon viele Funktionen eingebunden werden, die für Entwickler wichtig sind und viele Aspekte der Praxis abdecken. Sparx Systems fühlt sich der Weiterentwicklung sowohl des UML-Modellierungswerkzeuges Enterprise Architect als auch der zahlreichen Plug-Ins verpflichtet, die anwenderspezifische Anforderungen erfüllen.

Kontaktdaten

SparxSystems Software GmbH
Office: Handelskai 340/5, 1020 Vienna, Austria
Tel.: +43 (0)662 90 600 2041
Fax: +43 (0)662 90 333 3041
e-Mail: sales@sparxsystems.eu
Verkauf und Bestellabwicklung: sales@sparxsystems.eu
Produkt-Support: support@sparxsystems.eu
Internet Deutsch: www.sparxsystems.de Internet English: www.sparxsystems.eu Deutscher Blog: blog.sparxsystems.de
Englischer Blog: blog.sparxsystems.eu

