

Einsatz von Enterprise Architect

Von Sparx Systems

*Alle Unterlagen © Sparx Systems 2014 PTY
Übersetzt von SparxSystems Software GmbH*

<http://www.sparxsystems.de>

Inhaltsverzeichnis

Einführung	3
Einsatzmöglichkeiten.....	3
Wahl der Datenhaltungsart (Repository)	3
EAP-Datei-Repository.....	4
Firebird-Datei-Repository	4
DBMS-Repository.....	4
Einsatzvarianten.....	5
Einzelstandort.....	5
Mehrere Projekte	6
Mehrere Standorte	9
Cloud-Dienst (Verbindung mit HTTP).....	9
WAN Optimizer	11
RAS	11
Änderungsnachverfolgung.....	11
Auditing.....	12
Verwendung von Baselines	12
Version control / Package control	13
Reusable Asset Service.....	16
Zusammenarbeit im Team	16
MDG Technologies	16
Reference Data Export/Import	17
Security.....	17
Workflow.....	18
Model Forum (Team Review)	18
Model Mail	18
Performanceoptimierung	19
Remote Installation.....	20
Remote Installation von Enterprise Architect	20
Remote Floating License Installation.....	21
Anhang.....	22
Unicode Unterstützung für .eap-Dateien.....	22

Einführung

Sparx Systems Enterprise Architect ist für den Einsatz in großen Unternehmen konzipiert. Als skalierbare Modellierungsplattform bietet Enterprise Architect eine Reihe von Einsatzmöglichkeiten, um der Vielfalt moderner Unternehmen gerecht zu werden.

Dieses Weißbuch beschreibt die verfügbaren Optionen für den unternehmensweiten Austausch von Modellinformationen.

Typische Unternehmensstrukturen

Nachstehend sind typische Organisationsstrukturen aufgelistet, denen man bei der Bereitstellung einer Modellierungs-Software innerhalb von Firmen begegnet. Jede hat ihre eigenen Bedürfnisse und somit eine spezielle und optimale Software-Konfiguration.

- Einzelner Standort
 - Ein Gebäude mit vielen Nutzern über LAN
 - Mehrere Gebäude mit Cloud-Verbindung
- Einzelner Standort – mehrere Projekte
 - Mehrere Datenhaltungen (Repositories)
 - Ein Repository – mehrere Projekte
- Mehrere Standorte
 - Mehrere Standorte mit einem langsamen WAN oder einer Cloud-Verbindung
 - Firmenzentrale mit mehreren Vertragspartnern, die an ihren Standorten arbeiten

Einsatzmöglichkeiten

Enterprise Architect bietet eine Reihe von Funktionen zur Vereinfachung der Implementierung in großen organisatorischen Strukturen. Richtig kombiniert ergeben diese Funktionen die nötige Flexibilität, eine passende Konfiguration mit ausreichender Skalierbarkeit vorzusehen.

Diese Eigenschaften sind von der installierten Edition von Enterprise Architect und der eingesetzten Art der Datenhaltung abhängig. Die Datenhaltung des Modells wird auch als "Repository" bezeichnet.

Weitere Details über die Funktionen und Repository-Arten, die von verschiedenen Enterprise Architect Editionen unterstützt werden, sind auf der Seite [Vergleich der Editionen](#) zu finden.

Folgende Funktionen unterstützen die unternehmensweite Bereitstellung von Enterprise Architect:

- Datenhaltung: Repository-Art
- Zugriff: lokal versus Cloud-Dienst oder RAS
- Change Management
- Zusammenarbeit im Team

Wahl der Datenhaltungsart (Repository)

Bei allen Editionen oberhalb der Enterprise Architect Professional Edition können Modelle dateibasiert oder in einem DBMS (Datenbanksystem) gespeichert werden. Beide Varianten haben Vor- und Nachteile.

Die nachfolgende Tabelle vergleicht dateibasierte Speicherung mit einer Speicherung in einem DBMS:

Funktion	EAP	Firebird	DBMS
Replikation	ja	nein	nein
Dateibasiert	ja	ja	nein
Anzahl der Nutzer	1..~10	1	unlimitiert
Cloud Nutzer	nein	unlimitiert	unlimitiert
Korruptierbar	ja	ja	nein

Tabelle 1: Vergleich filebasiertes Repository mit DBMS-Repository

EAP-Datei-Repository

EAP Dateien (Enterprise Architect Project) basieren auf der Microsoft Jet 3.5 Datenbank (MS Access '97 format .mdb). Enterprise Architect unterstützt auch die Jet 4.0 Datenbank-Engine.

Hinweis: Das Jet 3.5 Format unterstützt Unicode nicht. Für Details zur Unterstützung von Unicode siehe Anhang [Unicode Unterstützung für .eap-Dateien](#) und die Hilfeseite [Basics](#).

Die Vorteile dieser Repository-Art sind:

- Die Repository-Replikations-Funktion mit Rückmischung und Konfliktbehebung
- Einfacher Dateizugriff über ein freigegebenes Netzlaufwerk

Die Grenzen, die bei dieser Repository-Art zu beachten sind:

- Gleichzeitiger Zugriff ist auf kleine Benutzergruppen beschränkt.
- In seltenen Fällen kann es zur Beschädigung von Daten kommen, falls ein Netzwerk- oder Stromausfall während der Bearbeitung auftritt.
- Der Jet Datenbanktreiber hat eine Größenbegrenzung. Empfehlung: unter 30-40MB

Firebird-Datei-Repository

Firebird-Dateien basieren auf der Interbase DBMS. Wie beim EAP-Repository ist dies dateibasierend und der Zugriff kann

- auf einem lokalen Laufwerk für Einzelnutzung oder
- über das Cloud-Service mit Mehrfachzugriff erfolgen.

Hinweis: Die Firebird-Datei erlaubt keinen Mehrfachzugriff auf einem Netzlaufwerk.

Firebird-Dateien sind wesentlich robuster als EAP-Dateien und sie unterstützen Unicode. Die bei EAP-Dateien unterstützte Replikationsfunktion gibt es bei Firebird-Dateien nicht.

DBMS-Repository

Mithilfe einer DBMS-Modell-Datenhaltung werden die Grenzen der dateibasierten Repositories überwunden. Für eine größere Nutzerbasis bieten dedizierte DBMS-Server schnellere Reaktionszeiten als die Jet-EAP-Dateien auf Netzlaufwerken. Darüber hinaus werden alle durch äußere Bedingungen verursachte Netzwerk-Aussetzer durch ein Rollback einer Transaktion durch den DBMS-Server automatisch berichtigt. DBMS-Repositories können auch über den Cloud-Dienst angesprochen werden. Eine Replika-Funktion steht nicht zur Verfügung, kann aber durch einen XMI-Merge ersetzt werden. Wenn der Cloud-Dienst verwendet wird, können alle Nutzer sehr einfach auf einem zentralen Repository arbeiten.

Enterprise Architect unterstützt 10 gängige Arten von DBMS-Server-Repositories. Details zu unterstützten DBMS-Arten können Sie auf der Hilfeseite [Server Based Repositories](#) nachlesen.

Einsatzvarianten

Enterprise Architect unterstützt vielfältige Einsatzvarianten. Da Enterprise Architect eine arbeitsstation-basierte (Client-)Applikation ist, kann das Programm auch ohne einen Server einfach genutzt werden. Für hohe Skalierbarkeit stehen serverbasierte Varianten zur Verfügung:

- Verbindung mit einem DBMS
- Lizenzverwaltungsdienst für Floating Licenses „Keystore Service“
- Verbindung über den Cloud-Dienst
- MDG-Technologies

Beim Einsatz von Enterprise Architect in einem Unternehmen können Sie eine Kombination obiger genannter Optionen verwenden. Die Verwendungsarten reichen von Nutzern, die in ihren persönlichen, filebasierten Repositories arbeiten, bis zu großen Nutzerteams an verschiedenen Standorten, die gemeinsam auf einem zentralen Repository arbeiten.

Im folgenden Abschnitt werden Einsatzschemas diskutiert, die auf den typischen Unternehmensstrukturen basieren, wie sie in der Einleitung skizziert wurden.

Hinweis: Informationen über die Remote (unbeaufsichtigte) Installation über ein Firmennetzwerk finden Sie im Anhang: Remote Installation von Enterprise Architect

Der nachfolgende Abschnitt diskutiert verschiedene Implementationsvarianten für typische Unternehmensstrukturen.

Einzelstandort

Das Szenario, das am häufigsten auftritt, ist ein einziger Einsatzort mit einem einzigen gemeinsamen Repository für ein bestimmtes Projekt oder auch für mehrere Projekte. In größeren Organisationen teilen sich Projekte normalerweise eine Reihe von Gemeinsamkeiten. Diese können aus gemeinsamen Datenbanken, gemeinsamen Treiberprogrammen oder gemeinsamen Reportvorlagen bestehen, die den Unternehmensrichtlinien entsprechen müssen.

Eine EAP-Datei auf einem gemeinsamen Laufwerk

Ein einfaches Setup. Die Repository-Datei wird auf ein Directory auf einem Fileserver gelegt, auf die von allen Nutzern über das lokale Netzwerk (LAN) zugegriffen werden kann. Wie schon erwähnt, gibt es Begrenzungen für .eap-filebasierte Repositories – bei bis zu 10 Nutzern sollte die Reaktionsgeschwindigkeit ausreichend sein.

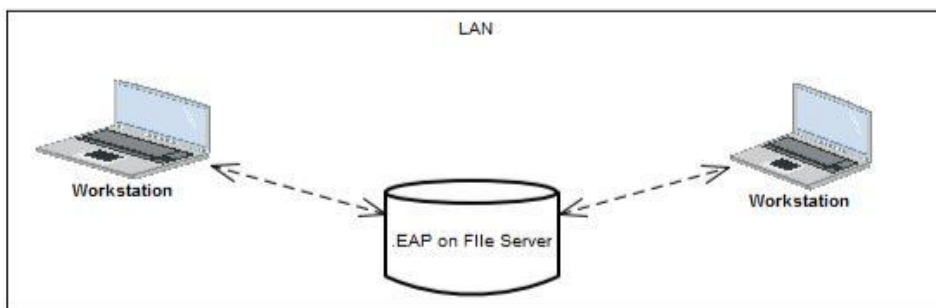


Abb. 1: Workstationzugriff auf ein dateibasiertes Repository

Verwendung eines DBMS-Repositories

Das DBMS-Repository bietet ein weitaus robusteres Umfeld. Es ist jedenfalls bei großen Nutzerzahlen erforderlich.

Für sehr große Repositories unterstützt das DBMS-Interface eine „Lazy Load“ Option. Wird ein Repository mit dieser Option geöffnet, werden vorerst nur Daten für den zunächst sichtbaren Baum im Project Browser geladen. Dies erlaubt ein schnelleres Öffnen des Projekts zum Preis von kurzen Verzögerungen, wenn im Project Browser weitere Baumteile aufgeklappt werden.

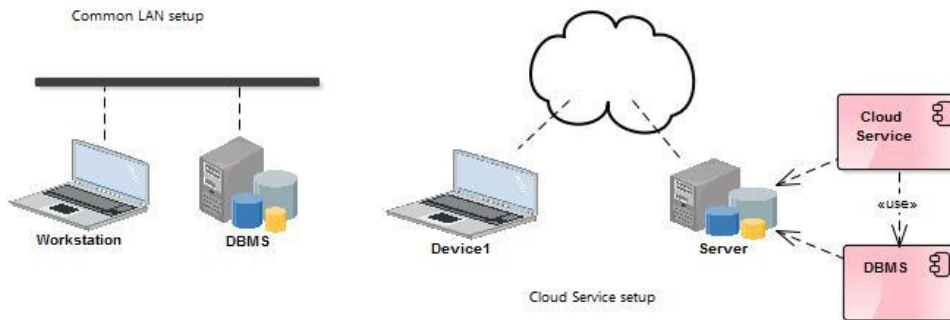


Abb. 2: LAN- und Cloud-Verbindung zu einem DBMS

Mehrere Projekte

Wenn mehrere Projekte gleichzeitig entwickelt werden, ist es möglich, dass Kern-Code, Skripte, Einstellungen, Templates und Reports existieren, die bei allen Projekten gleich sein sollen. Da die Größe des Enterprise Architect Modells nur durch die Parametrisierung des verwendeten DBMS begrenzt wird, ist die einfachste Lösung, alle Projekte in einem Repository zu halten und mit dem Rechtesystem den Schreibzugriff für die einzelnen Projektgruppen einzuschränken. Es kann aber auch Gründe geben, mehrere Repositories mit Datenaustausch vorzusehen. Die nachfolgenden Absätze diskutieren diese Varianten.

Ein Repository - Mehrere Projekte

Verwenden Sie entweder ein Model-Repository mit mehreren Projekt-Wurzel(Root)-Knoten oder mit mehreren View-Knoten (siehe nachfolgende Abbildung). Jeder Root-Knoten oder jede View enthält ein eigenständiges Projekt. Gemeinsamkeiten zwischen Projekten (Frameworks, Foundation Classes etc.) können auch unter einem eigenen Wurzel-Knoten oder einer eigenen View erfasst werden.

Ein einziges Repository zu verwenden, stellt sicher, dass gemeinsamer Code und andere gemeinsame Teile nur einmal gewartet und aus den verschiedenen Projekten zitiert werden können.

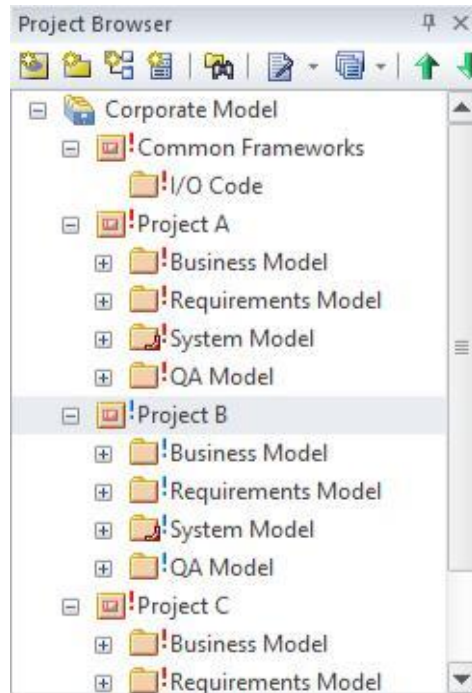


Abb. 3: Verschiedene Projekte mit gemeinsamem Framework

Darüber hinaus können in Enterprise Architect Sicherheitsfunktion verwendet werden, um den Zugriff eines Benutzers auf bestimmte Bereiche (Baumzweige) zu beschränken. Elemente außerhalb der Rechtezuweisungsbereiche können in Diagrammen zitiert (eingelinkt) werden, Datenänderungen an diesen zitierten Elementen bleiben aber jenen Nutzern vorbehalten, die ein Schreibrecht auf den Speicherort im Browser haben.

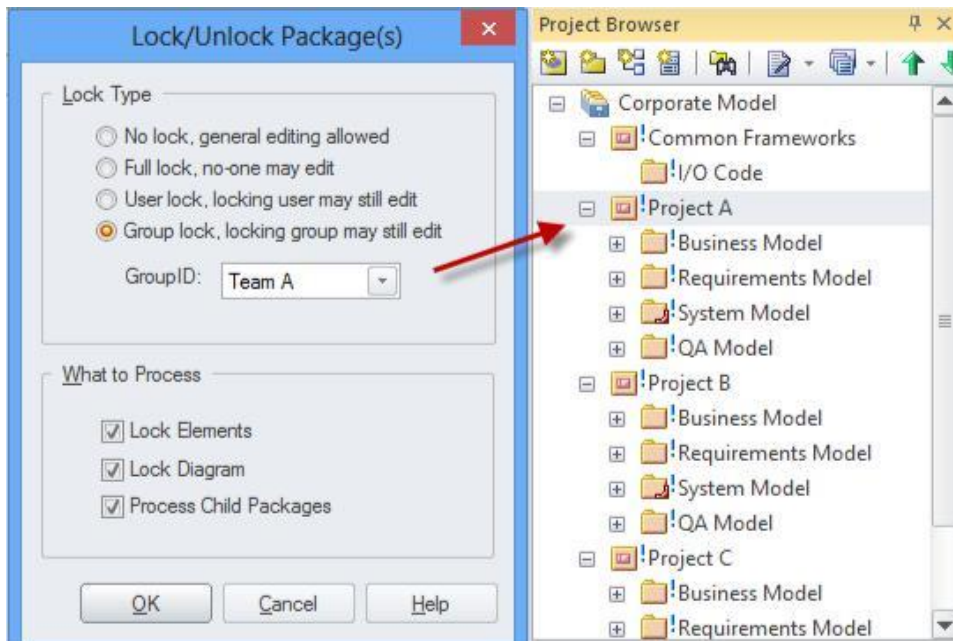


Abb. 4: Ansicht des Project Browsers mit Gruppen-Lock

Die nachfolgende Abbildung zeigt den Project-Browser, wie ihn ein Nutzer sieht, der in der Gruppe „Team B“ Mitglied ist. Ein ! zeigt dem Nutzer an, dass er eine Bearbeitungslaubnis hat.

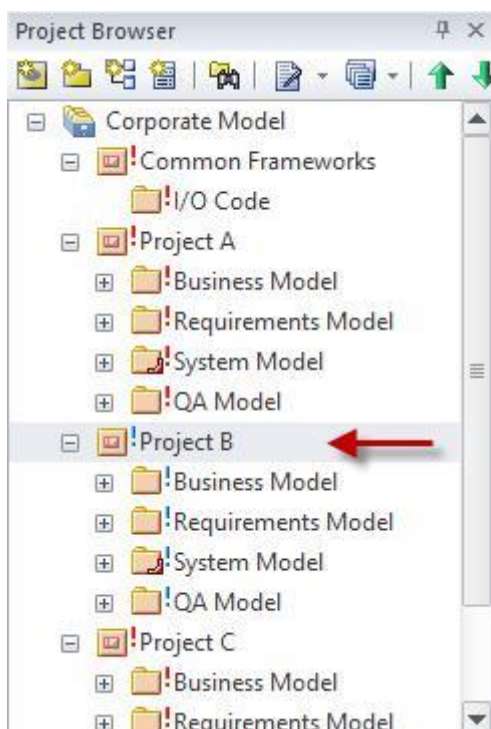


Abb. 5: Ansicht des Project Browsers für einen Nutzer des Teams B

Hinweis: Die Reportgenerierung lässt sich besonders einfach aufrufen, wenn alle Sub-Packages eines Projektes unter dem Projekt-Wurzelknoten angelegt werden. Dies ist auch für den späteren Bedarf einer „internen“ Projektversionierung wesentlich.

Mehrere Repositories – Austausch von Inhalten

Wenn Sie mehrere Projekte in einzelnen Repositories halten und Teillinhalte in diesen Projekten gleichartig verwenden müssen, können Sie verschiedene Funktionen zum Datenaustausch oder zum Ansprechen gemeinsamer Quellen verwenden. In all diesen Fällen wird empfohlen, ein Haupt-Repository zu verwenden, um die gemeinsamen Daten zu warten und zu verwalten.

In diesen Szenarien gibt es typisch zweierlei Arten von Daten:

- Packages, i. e. Model-Packages und gemeinsame Frameworks
- Ressource Daten wie Type-Definitionen und Reportvorlagen (Templates)

Es gibt mehrere Möglichkeiten zum Austausch dieser Datenarten zwischen Projekt-Repositories und dem Haupt-Repository.

Packages:

- Differenzbildung mittels Baselines (siehe Verwendung von Baselines)
- RAS (Reusable Asset Service)
- Package Control
- Externes Versionskontrollsystem
- XMI Export/Import

Ressource Daten

Zum Austausch von Ressource Daten gibt es zwei Mechanismen:

- **Reference Data Import/Export**
Die Reference Daten können periodisch vom Haupt-Repository aus mit der Funktion **Export Reference Data** aktualisiert werden
- **MDG Technologies**
Mit MDG Technologies können Sie ein Modul mit Ressourcen erstellen, das sich exportieren lässt. Dieses MDG Technologies Modul kann von jeder Instanz des Enterprise Architects auf einer Workstation referenziert und automatisch geladen werden. Weitere Details im Abschnitt MDG Technologies.

Große Projekte - Branching und Merging mehrerer Phasen

In großen Entwicklungsprojekten mit zyklischer Auslieferung kann die Vorbereitung der nächsten Phase ("Branch") in einer Repository-Kopie – einem "Zweig"-Repository - durchgeführt werden, während das Ausgangsmodell in der Entwicklung weiter verwendet wird. In beiden Richtungen – vom Original zum Zweig oder vom Zweig zum Original – kann durch Baselineerstellung und Differenzbildung mit „Load other Baseline“ selektiv gemischt werden. Dadurch können auch gemeinsame Teile aktualisiert werden. Es ist aber auch möglich – speziell bei SOA (Service Oriented Architecture) – einen Project Browserzweig im selben Repository zu duplizieren. Das kann mit „Copy Package to Clipboard“ und „Paste Package from Clipboard“ manuell ausgeführt werden. Sollen auch gleich Versionskennungen vorgesehen und Programmdateireferenzen berichtigt werden, nutzen Sie bitte die frei zugänglichen Skripte auf <http://sparxsystems.de/FAQ/>.

Gemeinsame Nutzung von Anlagen

Bei Szenarien mit vielen, in verschiedenen Organisationen verteilten Repositories kann der Dienst „Reusable Asset Service“ (RAS) zum Austausch von gemeinsam zu verwendenden Inhalten verwendet werden. Als Beispiel kann das gemeinsame Arbeiten von Organisationen an einem Industriestandard genannt werden. Dies erfordert einen zentralen Anlaufpunkt mit der Möglichkeit, neue Modelldaten beizutragen oder die letzte Version der bearbeiteten Spezifikation zu laden.

Weiter Information hierzu sind im Abschnitt Reusable Asset Service zu finden.

Mehrere Standorte

Es existieren viele Varianten, bei denen ein externer Zugriff auf ein lokales Netzwerk erforderlich wird. Das reicht von Unternehmen mit mehreren Standorten auf einem Kontinent bis zu interkontinentaler Zusammenarbeit, zumeist bei externer Entwicklung. Mehrere Funktionen unterstützen diese Anforderungen:

- Der Cloud-Dienst
- WAN Optimizer
- RAS Dienst

Cloud-Dienst (Verbindung mit HTTP)

Der Cloud-Dienst ist ein http Verbindungsdienst, der auf einem internen LAN basierten Server oder auf einem externen webbasierten Server aufgesetzt werden kann. Dies ist kein Hos-

ting Service, kann aber auf einem Hosting Service (z. B. Amazon oder Azure) aufgesetzt werden.

HTTPS SSL Verbindungen können für sichere Verbindungen für einzelne Nutzer oder Teams zum Verbinden zu einem gemeinsamen Repository benutzt werden.

Der Cloud-Dienst ist sowohl für WAN Verbindungen zwischen Lokationen in verschiedenen Gebäuden als auch für interkontinentalen Repositoryzugriff geeignet.

Die Vorteile eines http-Zugriffs über den Cloud-Dienst sind:

- Die Workstations (Nutzerrechner) benötigen keine ODBC-Einrichtung für den DBMS-Zugriff
- Die Verbindungsinformation ist einfach und kann leicht verteilt werden
- Unterstützt internen und externen Zugriff
- SSL für sichere Verbindungen wird unterstützt

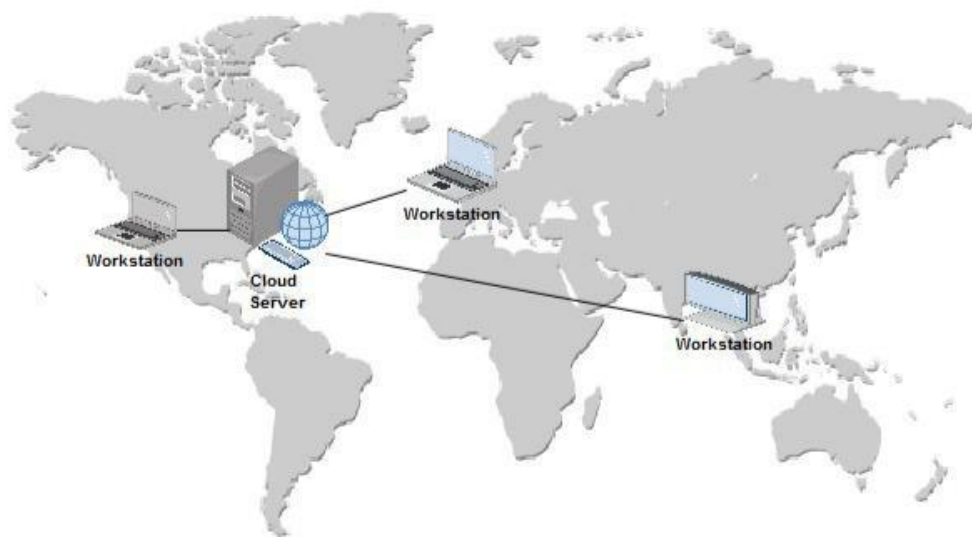


Abb. 6: Cloud-Dienst über Kontinente hinweg

Mittels Cloud-Dienst können Verbindungen zu mehreren Repositories mit unterschiedlichen Zugriffsrechten aufgesetzt werden. Jedes DBMS kann verschiedenen Cloud-Dienst-Verbindungen mit unterschiedlichen Zugangsrechten zugeordnet werden.

Die nachfolgende Abbildung zeigt die beteiligten Komponenten einer einfachen Konfiguration:

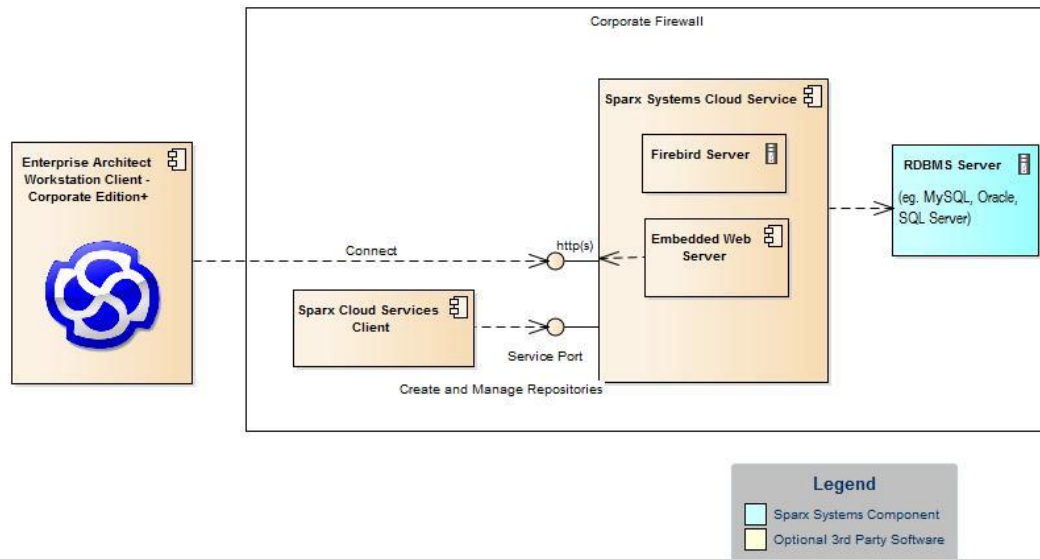


Abb. 7: Architektur des Cloud-Diensts

Hinweis: Der Cloud-Dienst ist ab der Corporate Edition verwendbar und im Lieferumfang enthalten.

Weitere Details zum Cloud-Dienst sind auf der Hilfeseite [Connecting to Projects Via the Cloud](#) verfügbar.

WAN Optimizer

Der WAN Optimizer stellt eine Datenkompression zwischen Arbeitsstation und DBMS Repository zur Verfügung. Der Cloud-Dienst enthält diese Datenkompression, aber der WAN Optimizer parallelisiert auch den ODBC-Zugriff auf das DBMS.

Für ein Szenario, das sehr strenge Sicherheit benötigt (HTTPS Verbindung ist nicht ausreichend), kann der WAN Optimizer als Alternative zur Verbindung über ein WAN eingesetzt werden.

Weitere Details finden Sie auf der Hilfeseite [WAN Optimizer](#).

RAS

Siehe Abschnitt Reusable Asset Service weiter unten.

Änderungsnachverfolgung

Enterprise Architect unterstützt mehrere Funktionen, um Änderungen an Modellen nachverfolgen zu können. Die einzelnen Funktionen dienen zu verschiedenen Zwecken, abhängig von Ihrer Modellgestaltung und Ihrem Entwicklungsvorgehen.

Die Nachverfolgungsfunktionen bestehen aus:

- Auditing (Protokollierung)
- Baselineing (Snapshot speichern, Differenzen einsehen)
- Versionskontrolle extern
- Reusable Asset Service (RAS)

Auditing

Die Auditfunktion dient zur Aufzeichnung von Änderungen in einem Modell. Aufgezeichnet wird, **wer** ein Element verändert hat, **wann** dies erfolgt ist und **was** geändert wurde. Sowohl der ursprüngliche Inhalt vor der Änderung als auch der neue Inhalt werden aufgezeichnet. Das kann speziell zum Aufzeichnen des Verlaufs von Anforderungsmodellen verwendet werden.

Die nachfolgende Abbildung zeigt die Darstellung einer Elementeänderung in der Audit View. Sichtbar sind mehrere Änderungen. Die erste Änderung wurde selektiert und die Details dazu sind im rechten Abschnitt zu sehen.

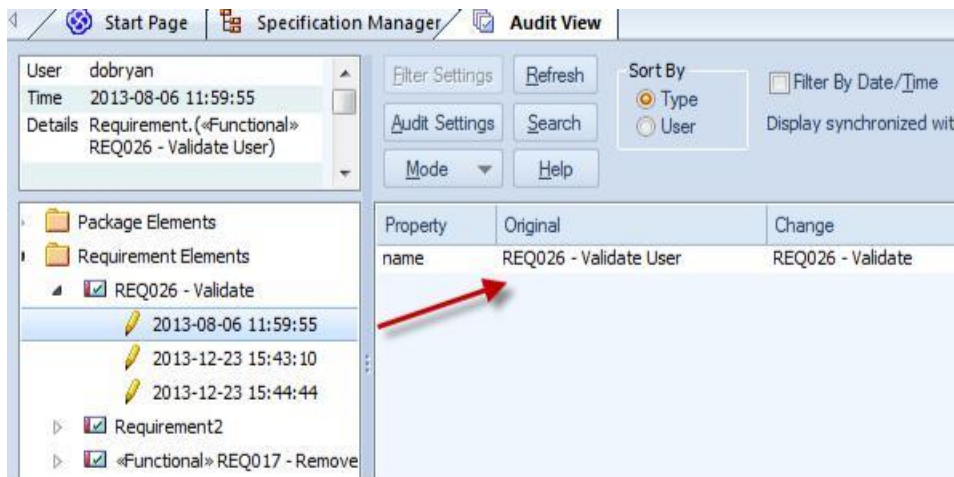


Abb. 8: Audit View - Änderung an einer Anforderung

Mit aktivierter Auditing View kann auch im Fenster View | System Output (auch zu öffnen mit Ctrl+Shift+8) die Änderungshistorie zum selektierten Element angezeigt werden. Es ist dabei egal, ob Sie das Element in einem Diagramm, im Project Browser, einer List View oder z. B. im Specification Manager selektieren.

Diese sinnvolle Funktion kann allerdings bei Aufzeichnung über lange Zeit oder bei einer sehr großen Anzahl an Nutzern das Repository (DBMS) stark belasten. Dies kann vermieden werden, indem Sie auch Baselines erstellen und periodisch ältere Inhalte des Audit Logs archivieren und im Repository löschen. Archivierte Audit Logs lassen sich bei Bedarf auch reimportieren. Baselines können gegen den jeweils aktuellen Modellstand verglichen werden und haben gegenüber dem Audit Log den Vorteil, dass einzelne Differenzen aufgehoben werden können oder dass gesamtheitlich auf den Stand der Baseline zurückgerollt werden kann.

Weitere Details finden Sie auf der Hilfeseite [Auditing](#).

Verwendung von Baselines

Die zuvor vorgestellte Audit Funktion bietet eine kontinuierliche Aufzeichnung der Änderungen an Elementen. Die Baseline Funktion erlaubt die einfrierende Abspeicherung des aktuellen Modellstandes eines Browserzweiges (Snapshot) mit der Möglichkeit, den später aktuellen Modellstand mit Baselines vergleichen zu können und Differenzen wieder aufheben zu können. Baselines können periodisch oder zu ablaufbedingten Zeitpunkten (Dokumentenerstellung, Variantenentwurf, ...) erstellt werden.

Branching mit Baselines

Baselines können auch für ein Branching mittels dupliziertem Repository verwendet werden. In beiden Richtungen (im Ausgangsrepository und im Branch-Repository) kann auf die Baselines des anderen Repositories zugegriffen, Differenz gebildet und eingemischt werden - mit der Funktion „Load other Baseline“.

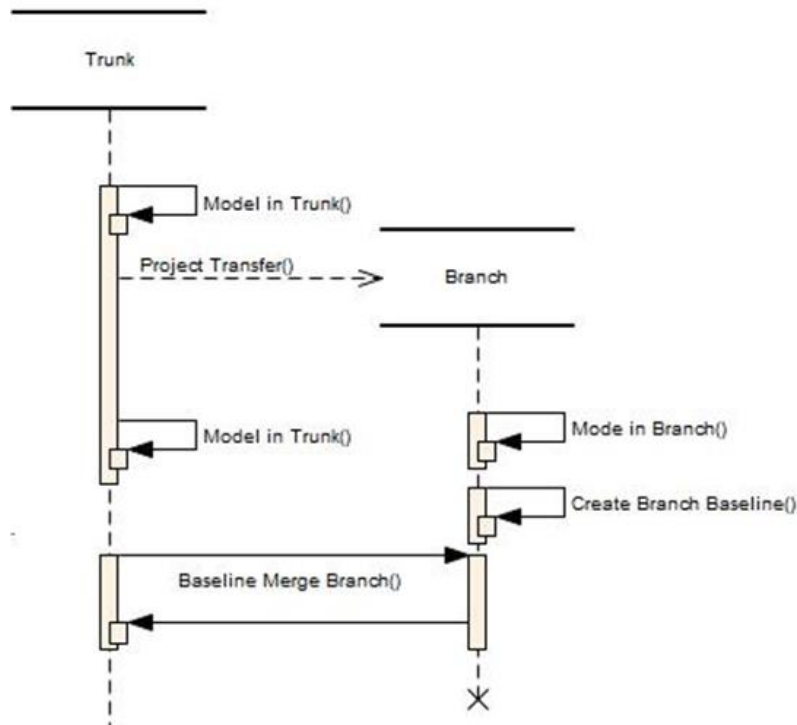


Abb. 9: Baseline zum Mischen zwischen Branch und Trunk

Weitere Informationen zum Erstellen von Baselines und zum Abruf von Differenzen finden Sie unter [Package Baselines](#) auf der Hilfeseite.

Version control / Package control

Enterprise Architect unterstützt die Versionskontrolle von Packages einschließlich darin enthaltener Subpackages in einem zentralen Versionskontrollsystem von anderen Herstellern. Folgende Vorteile entstehen:

- Aufzeichnung der Historie von Enterprise Architect Packages, einschließlich der Möglichkeit, vorausgehende Versionen wiederherzustellen.
- Abgleich von Packages zwischen verschiedenen Nutzern mit getrennten Repositories

Diese Versionskontrollfunktion kann mit Versionskontrollsystemen verwendet werden, die folgende kompatible Standards unterstützen:

- SCC Standard
- CVS
- Subversion
- Microsoft TFS

Wann sollte ein Versionskontrollsystem verwendet werden?

Auch wenn die Versionskontrolle bei der Programmerstellung als Standard für Codedateien verwendet wird, ist zu bedenken, dass die Versionierung von Modellen nicht so einfach zu im-

plementieren ist. Der wesentliche Unterschied liegt darin, dass ein Programm eine einfache Textdatei ohne explizite Referenzen nach außen ist, während Modellinhalte eines Browserzweiges viele Verlinkungen zu anderen Teilen des Modells aufweisen. Diese Eigenschaft schließt die Anwendung von speziellen Funktionen der Versionierungssysteme, die bei der Codeversionierung verwendet werden können, aus. Ein mehrfaches, überlappendes Auschecken derselben Codedatei durch mehrere Programmierer und eine automatische Mischung beim Einchecken sind nur bei Codedateien denkbar. Modellteile mehrfach auszuchecken und dann wieder zu mischen, ist wegen der Verkettungen nicht möglich.

Dieser Abschnitt stellt keine grundsätzliche Empfehlung dar, Versionskontrolle nicht zu verwenden, sondern ist der Rat, die im Enterprise Architect selbst enthaltenen Funktionen zu erkunden, bevor Sie sich für das wesentlich komplexere Szenario mit einem Versionskontrollsystem entscheiden.

Vorteile:

- Einfach, wenn die regelmäßige Nachverfolgung von Änderungen erforderlich ist
- Versionen können angezeigt und zurückgerollt werden
- Ein Versionspaket kann auch ein einzelnes Package sein (ohne Sub-Packages)
- Versionskontrolle kann zum Abgleich zwischen mehreren Standorten mit eigenen Repositories verwendet werden
- Die zwingende Package-Sperrung kann genutzt werden, um Eingabekonflikte zu verhindern

Nachteile:

- Die Versionsdaten werden extern gespeichert; sie können verloren gehen, während Baselines im Repository gespeichert werden
- Wegen des Verarbeitungsaufwandes und der damit verbundenen Laufzeit mit dem Versionskontrollsystem-Interface muss die Granularität der Versionierungspakete klein gehalten werden; das wird Komplikationen mit Verkettungen nach außen verursachen.
- Die Sperre durch das Auschecken kann den Modellzugriff blockieren, speziell, wenn ein anderer Nutzer vergisst, das Package wieder einzuchecken – **es kann ja immer nur ein Versionskontroll-Nutzer in einem ausgecheckten Package zugleich arbeiten.**
- Bei kontinentübergreifendem Arbeiten kann die Interaktion mit einem Versionskontrollsystem sehr langsam werden.
- Bei Nutzung mehrerer Repositories werden nur die Package-Daten versionskontrolliert. Reference Data wie zum Beispiel Dokumenten-Templates, Kalendereinträge, Tagged Value- und Stereotypdefinitionen und das Glossar müssen manuell über Export/Import in alle Repositories transferiert werden.
- **Wenn Nutzer sehr oft Packages auschecken, kann das die Effizienz des gemeinsamen Arbeitens wegen der damit verbundenen Bearbeitungssperre stark senken.**
- Die Verwendung von externen Versionierungssystemen erfordert eine Konfiguration für jeden einzelnen Nutzer, ein eher komplizierter Prozess, bei dem Fehler zu Datenverlust oder zur Korruption von Daten führen kann.

Alternativen:

Als Alternativen bieten sich die voraus vorgestellten Funktionen Auditing, Baselines und RAS an. Sie können nach Bedarf kombiniert werden, um ein der Anwendung eines Versionskontrollsystems gleichwertiges Resultat zu erreichen.

Die nachfolgend genannten Punkte listen die alternativen Enterprise Architect internen Funktionen im Vergleich zur externen Versionskontrolle auf.

Gemeinsame Nutzung eines Repositories

- Cloud:
 - Der Cloud-Dienst erlaubt standortübergreifende Zugang, kein Bedarf für mehrfache Datenhaltungen

- Reference Data wie Dokumenten Templates, Glossar müssen nicht ausgetauscht werden.
- RAS:
 - RAS, bereitgestellt vom Cloud-Dienst, gestattet kontinentübergreifenden Austausch von Modellen und eine versionsbasierte Historie von Inhalten.
- Baselines:
 - Baselines können für den Austausch zwischen mehreren Repositories verwendet werden.
 - Baselines erlauben den Abgleich einzelner Differenzen – Merge-Feature.
 - Baselines erlauben Trunk- und Branchzyklen der Repository-Inhalte; Versionskontrolle unterstützt dies nicht.

Freigabeprozesse

- Auditing:
 - Auditing stellt eine einfache Liste der Änderungen für historische Einsicht bereit.
- Baselines:
 - Baselines erlauben die periodische Abspeicherung von Modellständen.
 - Differenzen können grafisch und textuell angezeigt werden.

Eingabesperre für Packages, Diagramme und Elemente

- Security:
 - Sperre eines Abschnitts gegen weitere Eingaben ist möglich (z. B. Einfrieren nach Dokumentenerstellung)
 - Die Sperrgranularität ist feiner als bei Verwendung eines Versionskontrollsystems; einzelne Elemente können für Nutzer oder Nutzergruppen gesperrt werden, unabhängig von Diagramm-Bearbeitungsrechten.
 - Diagramme können bei Bearbeitungsbeginn automatisch auf exklusives Bearbeitungsrecht gesetzt werden und bei Abspeicherung wieder automatisch für neuerliche Bearbeitung durch andere Nutzer freigegeben werden.

Zusammenfassend: Ist eine Historie der Änderungen erforderlich, wird empfohlen, die Vor- und Nachteile der Alternativen abzuwägen, bevor zur Anwendung eines externen Versionsierungssystems übergegangen wird.

Hinweis: Wird ein Versionskontrollsystem zum Abgleich zwischen vielen einzelnen Repositories verwendet, ergeben sich niedrige Nutzeranzahlen an den einzelnen Repositories und das könnte den Einsatz von .eap-Dateien nahelegen. Dies ist aber für große .eap-Dateien nicht empfehlenswert! .eap-Dateien sind nicht transaktionssicher und können korrumpieren, was dann auch die Inhalte im Versionskontrollsystem korrumpiert.

Für das Aufsetzen von Versionskontrollsystemen im Enterprise Architect finden Sie Anleitungen auf der Hilfeseite [Version Control](#).

Für die Verwendung von Versionskontrollsystemen schlagen Sie bitte im Weißbuch [Version Control Best Practices](#) nach.

Reusable Asset Service

Das Reusable Asset Service (RAS) unterstützt einen Verteilungsmechanismus wiederverwendbarer Modellstrukturen über ein Cloud-Dienst basiertes Repository in offener, einheitlicher und rascher Weise. Diese Modelle werden von zugeordneten Administratoren verwaltet, die anderen Cloud Dienst Nutzern die Möglichkeit zum Herunterladen einräumen.

Das wird für grenzüberschreitende und kontinentüberschreitende Organisationen sinnvoll sein, die gemeinsame Inhalte und Strukturen austauschen wollen, ohne ein komplexes, globales Versionskontrollsystem aufsetzen zu müssen.

Ein gutes Beispiel hierfür könnte die Arbeit von Normungsgremien sein. Die Autoren dieser Normen können Mitarbeiter verschiedener Organisationen sein, an verschiedenen Standorten, vielleicht auch verschiedenen Kontinenten und sie werden Strukturen bei der Entwicklung ihrer globalen Standards austauschen wollen. Anstatt manuell die Modelle oder XMI-Datei zu verteilen, können die verantwortlichen Architekten den Cloud-Dienst mit dem Reusable Asset Service zur Zusammenarbeit verwenden und das Profil des Standards aktualisieren und zum Review oder zur Verwendung publizieren.

Weitere Information zur Verwendung des Cloud-Dienstes finden Sie auf der Hilfeseite [Reusable Asset Service](#).

Zusammenarbeit im Team

Bei Arbeiten in interdisziplinären Teams sind Zusammenarbeitsfunktionen für Inhalte und Ideen wesentlich. Enterprise Architect stellt mehrere Zusammenarbeitsfunktionen zur Verfügung: ein repositorybasiertes Forum, ein Mailingsystem und Funktionen zur Verteilung von Ressourcen wie zum Beispiel von Report Templates (MDG Technologies). Beim Ausrollen von Enterprise Architect wird eine Kenntnis über die Arbeitsweise dieser Funktionen hilfreich sein. Hinweis: Einige dieser Funktionen setzen eine Aktivierung der Securityfunktion voraus.

MDG Technologies

In großen Organisationen können mehrere Teams verschiedene Modelle oder Anwendungen entwickeln, bei denen dieselben Grundlagen angewandt werden sollen. Diese Grundlagen reichen von firmenspezifischen Profilen über Codetemplates bis zu Reportvorlagen mit firmenspezifischer Formatierung.

Mit MDG Technologies kann man gemeinsame Grundlagen (Ressourcen) erstellen, die von Gruppen mit verschiedenen Repositories in der Organisation verwendet werden. Diese gemeinsamen Ressourcen umfassen:

- Profile (um Modellierungssprachen zu definieren oder zu modifizieren)
- Pattern (wiederverwendbare Modellstrukturen)
- Tagged Value Types (für anwenderdefinierte Zusatzfelder)
- Code Module (Code Generation Templates)
- MDA Transformationen
- Report Templates
- Linked Document Templates
- Bilder
- Skripte
- Maskenanordnungen
- Model Views
- Modellsuchen
- Modellvorlagen

Einmal angelegt, kann ein MDG-Technology-Modul auf einem gemeinsam erreichbaren Laufwerk abgelegt und durch Einstellungen des Enterprise Architects auf der Arbeitsstation referenziert werden, wodurch es beim Programmstart automatisch eingebunden wird.

Weitere Details zur Einrichtung finden Sie in der Hilfeseite [MDG Technology SDK](#).

Reference Data Export/Import

Eine einfache Alternative Technologies zur Verteilung der Ressourcen verglichen mit MDG stellt der Reference Data Export/Import dar. Typischerweise wird ein Hauptrepository aktuell gehalten und die Reference Data werden periodisch an andere Repositories verteilt.

Typische Teile, die von einem Repository aus verteilt werden, sind:

- Glossar
- Type definitions (Statusliste, ...)
- Resources, Clients
- RTF und HTML Templates
- Security

Beim Export von Reference Data wird eine standard-XML-Datei erzeugt. Diese enthält die Tabellenbenennung, die gefilterten Inhalte, Zeilen und Spalten.

Weitere Details zum Export finden Sie auf der Hilfeseite [Sharing Reference Data](#).

Security

Obwohl die Security Funktion eine Option ist, die für kleine Gruppen nicht erforderlich ist, bietet ihre Verwendung auch für kleine Gruppen einige Vorteile. Diese reichen vom exklusiven Zugriffsrecht für den einzelnen Nutzer bis zu komplexen Funktionen im Mehrnutzerbetrieb.

Einige der Kernfunktionen der Security sind:

- Workflow Scripting
- Project Management, Gantt Diagramme und Kalender
- Model Mail
- Audit Tracking (Protokollierung mit Anmeldezuordnung)

Die Enterprise Architect Corporate Edition und die Bündeleditionen bieten eine anwenderspezifische Sicherheitsfunktion – Einschränkungsmöglichkeiten für das Ändern und Ergänzen des Modells. Ein Anmeldepasswort oder eine automatische Anmeldung mit der Windows-Login-Kennung ist erforderlich und Elemente können für einen Nutzer oder eine Nutzergruppe gesperrt werden.

Die Security Funktion in Enterprise Architect ist nicht zur Einschränkung gegen unautorisierten Zugriff auf einzelne Teile gedacht; vielmehr dient sie dazu, das gemeinsame Arbeiten und Entwickeln gegen kollidierendes Eingeben oder Ändern zu schützen und gegen Eingaben von Nutzern zu schützen, die nicht als Modellautoren vorgesehen sind. Eine einfrierende Sperrung für verbindliche Modellteile kann damit auch erfolgen.

Enterprise Architect unterstützt zwei Security-Varianten:

Standard Security Modus

Alle Elemente und Diagramme könnten unversperrt sein und ein Nutzer, abhängig von seinen Rechten, kann Elemente oder einen Satz von Elementen oder Diagramme für sich oder für eine Gruppe sperren, in der der Nutzer Mitglied ist.

Rigorous Security Modus

Bei diesem Modus ist alles grundsätzlich schreibgeschützt außer ein Nutzer setzt für sich selbst eine Sperre auf ein oder mehrere Elemente oder Diagramme.

Die Security Funktion erlaubt, Elemente gegen Änderung durch Nutzer zu schützen, die nicht Sperreninhaber sind. Ein Einfügen der gesperrten Elemente in Diagramme ist aber für beliebige Nutzer dennoch möglich, wenn sie für diese Diagramme Schreibrecht besitzen. Die Elemente werden angezeigt, können selbst aber nicht verändert werden. Ein Beispiel: Die Definition eines Servers ist durch einen Architekten gesperrt, kann aber durch den Planer der Einsatzkonfiguration in ein Diagramm eingefügt werden. Auch eine Einzeichnung einer neuen Instanz ist möglich.

Weitere Details zur Security Funktion finden Sie auf der Hilfeseite [User Security](#).

Workflow

In einer modellgetriebenen Entwicklungsumgebung können viele Arbeitsabläufe im Design und während der Durchführung eines Projekts auftreten - z. B. Freigabeprozesse.

Durch Anwendung des Workflow-Scriptings im Enterprise Architect können Sie die Art und Weise des möglichen Vorgehens von Teammitgliedern vorgeben und somit einen besicherten Prozessablauf bei der Handhabung des Modells sicherstellen.

Das Einrichten von Workflow Skripten wird üblicherweise durch die Prozess-Management-Verantwortlichen, den für das Design und den Entwicklungsprozess Verantwortlichen (Qualitätssicherungsbeauftragten) durchgeführt. Als Administrator können Sie derart Ihren firmenspezifischen Arbeitsprozess sicherstellen.

Ein Weißbuch und ein Beispielmodell für das Workflow-Scripting finden sich auf [Workflow Scripting in Enterprise Architect](#) im Community-Bereich.

Weitere Details zum Workflow Scripting finden Sie in der Hilfeseite [Workflow Scripts](#).

Model Forum (Team Review)

Mit der Team-Review-Funktion von Enterprise Architect können Modellierer die Entwicklung und den Fortschritt eines Projektes diskutieren. Teammitglieder können Kommentare direkt im Modellierungswerkzeug veröffentlichen und einsehen und dabei ihre Kommentare mit Modellelementen verlinken. Bei verteilten Umgebungen können Nutzer ihr Enterprise Architect Modell mit einem Team-Review-Abschnitt eines anderen Modell-Repositories verbinden. Weitere Details hierzu finden Sie auf der Hilfeseite [Team Review Tools](#).

Werden mehrere Repositories in einem Projekt verwendet, können Team Reviews in anderen Repositories angesprochen werden. Weitere Details hierzu bietet die Hilfeseite [Team Review Connections](#).

Model Mail

In einem großen Entwicklerteam kann ein erheblicher Abstimmungsbedarf bestehen. Dabei ist es wesentlich, dass Nachrichten Modellelemente referenzieren können. Enterprise Architect stellt ein internes Mailsystem zum Austausch von Nachrichten zwischen am Modell arbeitenden Nutzern zur Verfügung.

Die wesentlichen Gründe, die Model Mail zu nutzen sind:

- Nachrichten der Model-Mail-Funktion erlauben eine Verlinkung auf Modelldetails (Diagramme, Elemente) – das ist in einem Standardmailsystem nicht möglich.
- Model Mail verhindert zahlreiche Messages in Standardmailsystemen, die stören könnten.
- Eine Kritik erreicht nur den Adressaten und ist im Modell nicht öffentlich.

Hinweis: Nachrichten an andere Mailsysteme werden mittels Automation Interface unterstützt, falls Sie dies benötigen. Ein Skript zur Kommunikation mit Outlook finden Sie im Weißbuch [Workflow Scripting in Enterprise Architect](#).

Performanceoptimierung

Beim Teilen großer Modelle über ein Netzwerk (LAN, WAN oder Cloud) sollten einige Überlegungen zur Performanceoptimierung angestellt werden. Grundsätzlich hängt die Performance von folgenden Punkten ab:

Von der verwendeten Repository-Art (EAP, FEAP, oder remote DBMS)

Die Netzwerk-Reaktionszeit hängt ab von:

1. Netzwerklast
2. WAN-Latenz (ein WAN mit niedriger Latenz hat eine Latenz unter 40-50ms)

Die nachfolgende Tabelle gibt Auskunft, welche Repository-Art für hohe Performance ausschlaggebend ist.

Repository-Art	Netzwerktyp	HTTP*	Nutzer	Optimal	Kommentar
EAP	Lokal	Nicht verfügbar	1-10	JA	Dateizugriff auf der Arbeitsstation
EAP	LAN	Nicht verfügbar	1-10	JA	Performance hängt von der Netzwerklast ab. Erfordert ein gemeinsames Laufwerk und Verzeichnis.
EAP	WAN	Nicht verfügbar	1-10	NEIN	Nicht empfohlen
EAP Replikation	Lokal	Nicht verfügbar		JA	Hilfreich für Off-Line-Arbeiten. Kann durch Cloud-Verbindung ersetzt werden.
FEAP	Lokal	X	1	JA	Keine ODBC-Verbindung. Mehrfach-Nutzerzugriff wird nicht unterstützt
FEAP	LAN/WAN/Web	JA	∞ *	JA	Cloud-Dienst erforderlich
DBMS	LAN	X	∞ *	JA	Gut für große Teams. Arbeitsstationen benötigen ODBC-Einrichtung
DBMS	WAN/Web	JA	∞ *	JA	Optimal für WAN/Web-Verbindung. Arbeitsstationen benötigen keine ODBC-Einrichtung. (MySQL & SQL Server verursachen weniger Last als Oracle!)

* Die maximale Anzahl gleichzeitiger Nutzer hängt von der physischen Kapazität der Datenbank und der Konfiguration des benutzten Servers ab.

Remote Installation

Remote Installation von Enterprise Architect

Zum Verteilen von Enterprise Architect über ein Netzwerk gibt es mehrere Anwendungen und Methoden. Die Verteilung kann mit Windows-Server-Werkzeugen oder mit Programmen wie dem Microsoft SCCM/SMS erfolgen. Dieser Abschnitt gibt eine kurze Einführung.

Übersicht

Das Installationsprogramm wird als MSI Installationspaket ausgeliefert. Mit dieser Datei können Sie die Installation über Windows Server oder SMS ausführen.

Falls Sie Dokumentation zu den Optionen der Windows MSI Installation benötigen: [Msiexec \(command-line options\)](#).

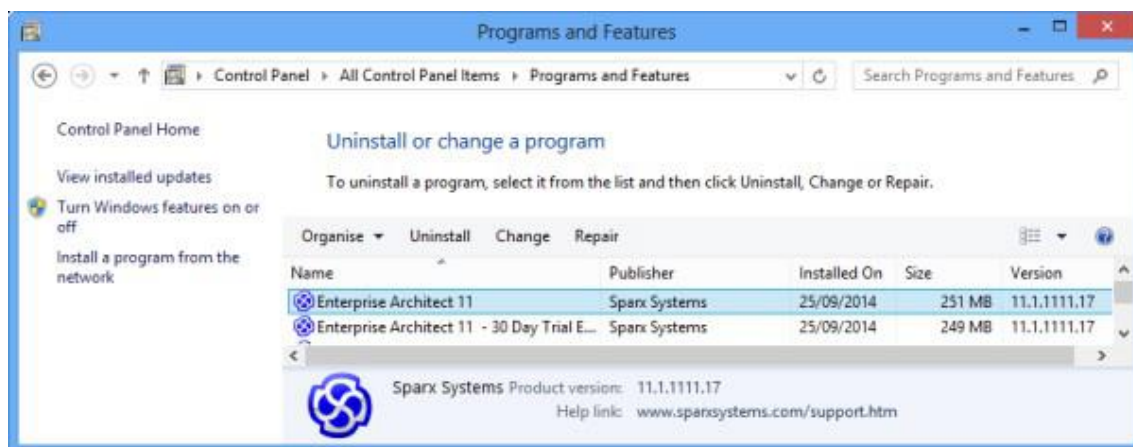
Hinweis: Laut Standardeinstellung installiert das Installationsprogramm den Enterprise Architect für den aktuellen Nutzer. Um für alle Nutzer zu installieren, geben Sie bitte an:

```
msiexec /i c:\easetupfull.msi /q allusers=2
```

Überprüfung nach der Installation:

Bitte beachten: Dieses Beispiel setzt voraus, dass für alle Nutzer installiert wurde.

1. Nach der Installation – wie nachfolgend abgebildet – mit **Windows | Add/Remove Programs** überprüfen, ob Enterprise Architect korrekt installiert wurde.



2. Überprüfen Sie bitte im „All Users“ Profil für den Desktop und im Start-Menü, ob Enterprise Architect erfolgreich für alle Nutzer installiert wurde. Abhängig vom Betriebssystem treten folgende Verzeichnisse auf:

Windows XP: C:\Documents and settings\All Users\Desktop

Windows 7+: C:\Users\Public\Desktop

Remote Floating License Installation

Während der automatischen Installation von Enterprise Architect können Registrierungseinträge für jeden Nutzer gesetzt werden, die ihm Floating Lizenzen ohne weitere Nutzerabfrage zuordnen. Die Registrierungseinträge unterscheiden sich für die dateibasierte oder die dienstbasierte Lizenzverwaltung:

1. Registrierungseinträge für dateibasierte Lizenzverwaltung:
 [HKEY_CURRENT_USER\Software\Sparx Systems\EA400\EA\OPTIONS]
 "SKT"=dword:00000000
 "SharedKeyFolder"="Y:\\Dev\\Licenses"
 "AutoCheckoutEx"=hex:1a,00,00,00
2. Registrierungseinträge für dienstbasierte Lizenzverwaltung:
 [HKEY_CURRENT_USER\Software\Sparx Systems\EA400\EA\OPTIONS]
 "SKT"=dword:00000001
 "SSKAddress"="sks://pathToKeystoreService"
 "SSKSPassword"="service password (encrypted)"
 "AutoCheckoutEx"=hex:1a,00,00,00

Bedeutung der Schlüssel

<u>SKT</u> (Shared Keystore Type)	Spezifiziert die Art der Lizenzverwaltung. Erlaubte Werte sind 0x00 für dateibasiert und 0x01 für dienstbasiert
<u>SharedKeyFolder</u>	Dieser Wert sollte auf das Verzeichnis oder den Netzwerkpfad mit der sskey.dat Datei zeigen. Wird nur verwendet, wenn SKT den Wert 0x00 hat. Das obige Beispiel zeigt auf ein Verzeichnis auf einem Netzlaufwerk: Y:\\Dev\\Licenses Hinweis: <ul style="list-style-type: none"> • Achten Sie bitte auf die doppelten Backslashes im Registrierungseintrag • Ein voller UNC-Pfad wird empfohlen, z. B.: „\\DevelopmentServer\EA Licenses” • Nutzer von Enterprise Architect müssen auf dieser Datei Les- und Schreibrechte haben • Wenn Sie eine Lizenzdatei auf einem Novell Server verwenden, ist die Groß-/Kleinschreibung exakt einzuhalten
<u>SSKAddress</u>	Die sks-Adresse zum Shared-Keystore-Service Endpunkt. Diese Einstellung wird nur verwendet, fall der SKT-Schlüssel einen Wert von 0x01 hat
<u>SSKSPassword</u>	Wenn der Shared-Keystore-Dienst ein Schlüsselwort erfordert, kann es in diesen Wert eingesetzt werden. Bitte beachten, dass dieser Wert verschlüsselt wird und nicht als Klartext eingegeben werden kann. Dieser Schlüssel wird nur verwendet, wenn der Schlüssel SKT den Wert 0x01 hat.
<u>AutoCheckoutEx</u>	Gibt an, welche Lizenzschlüssel beim Start von Enterprise Architect automatisch entnommen werden sollen. Jeder Schlüssel wird durch 4 Bytes deklariert, z. B.: 1a 00 00 00 Wobei die Bytes 1-2 der Lizenzcode sind (1a 00) und die Bytes 3-4 die Lizenzart bedeuten (00 00). Die zulässigen Werte zeigt die nachfolgende Tabelle.

Lizencodierungen für AutoCheckoutEx:

Lizenz	Code
Enterprise Architect Corporate	0200
Enterprise Architect Ultimate	1a00
Enterprise Architect Business & Software Engineering	1800
Enterprise Architect Systems Engineering	1900
MDG Integration for Visual Studio	0a00
MDG Integration for Eclipse	1400
MDG Link for Visual Studio	0300
MDG Link for Eclipse	0800
MDG Link for Doors	0e00
MDG Technology for SysML	1000
MDG Technology for DDS	1200
MDG Technology for Zachman Framework	1600
MDG Technology for TOGAF	1d00
MDG Technology for UPDM	1b00
RaQuest	0c00

Lizenztype für AutoCheckoutEx:
Standardlizenz: 0000
Akademische Lizenz: 0100

Beispiel: Angenommen alle Benutzer sollen sowohl eine Corporate-Lizenz als auch eine Visual-Studio-Integrationslizenz bekommen, dann wäre der Registry-Schlüsselwert:

"AutoCheckoutSharedKeyArray"=**hex:02,00,00,00,0a,00,00,00**

Anhang

Unicode Unterstützung für .eap-Dateien

Standardgemäß ist das .eap-Datei-Format das Jet 3.5 Format (Access 97). Dieses unterstützt keine Unicode-Zeichen. Um Unicode verwenden zu können, müssen Sie das Jet 4.0 Dateiformat verwenden (Access 2000).

Um die Unicodeunterstützung unter Jet 4.0 zu aktivieren, müssen Sie:

- 1) Im Hauptmenü anwählen:
Tools | Options | Genral
- 2) Die Option aktivieren: [x] Use Jet 4.0
- 3) Die Jet 4.0-Version der EABase.eap Datei herunterladen von:
http://www.sparxsystems.com/bin/EABase_JET4.zip
- 4) Die heruntergeladene Datei entpacken
- 5) Ausführen: **Project | Data Management | Project Transfer**
- 6) Mit **Project Transfer** im Dialog einen Datei- zu Dateitransfer ausführen mit
 - **Source:** Ihre Projektdatei (darf nicht mit EA geöffnet sein)
 - **Target:** Die heruntergeladene, entpackte Datei (darf nicht mit EA geöffnet sein)
- 7) Ab dem Öffnen der Jet 4.0 Datei sollten Sie nun die gewünschten Unicodezeichen eingeben und ohne Datenverlust speichern können.