

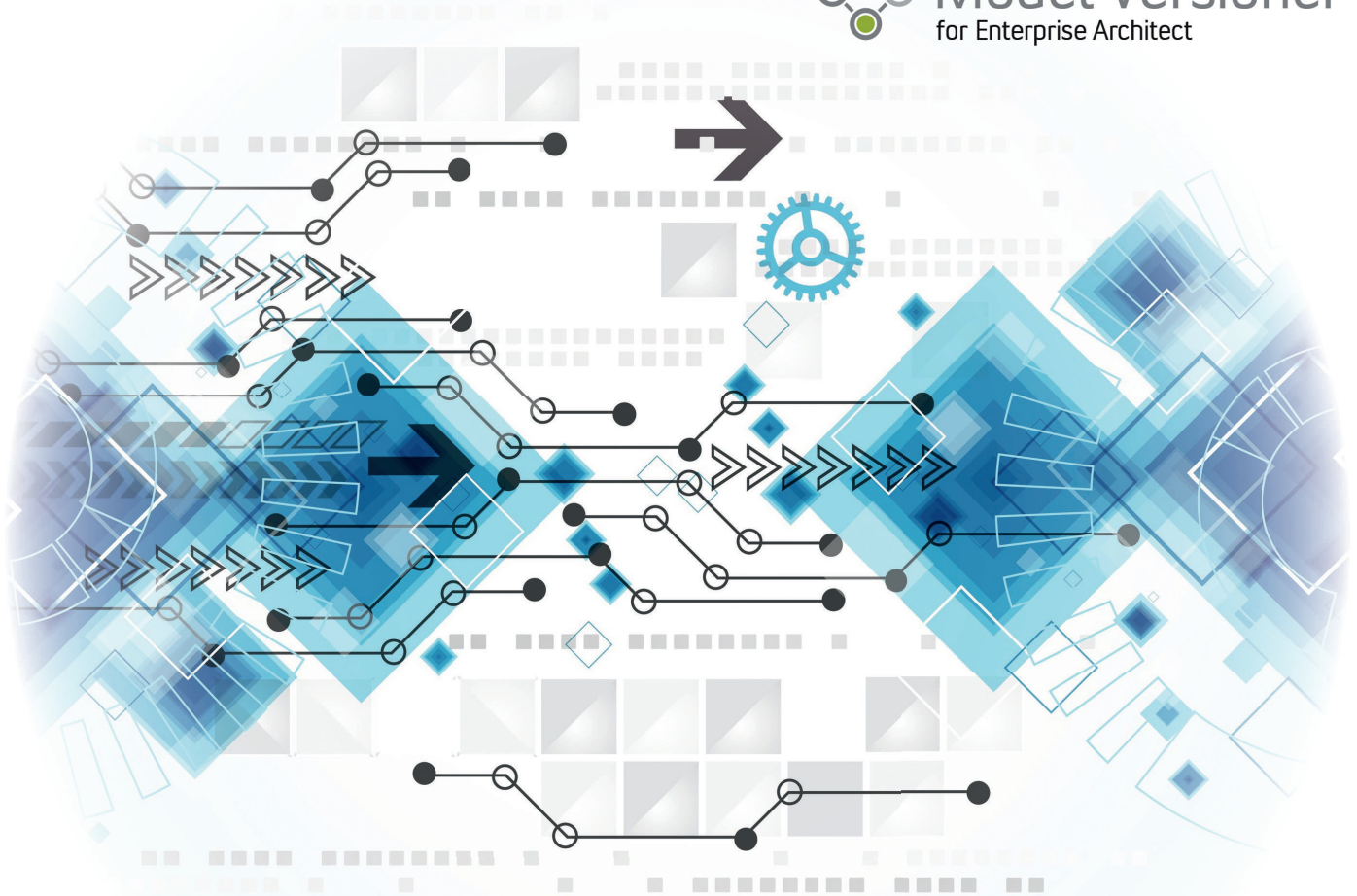
ELEKTRONIK PRAXIS

www.elektronikpraxis.de

Wissen.
Impulse.
Kontakte.

Februar 2016

LieberLieber[®]
Model Versioner
for Enterprise Architect

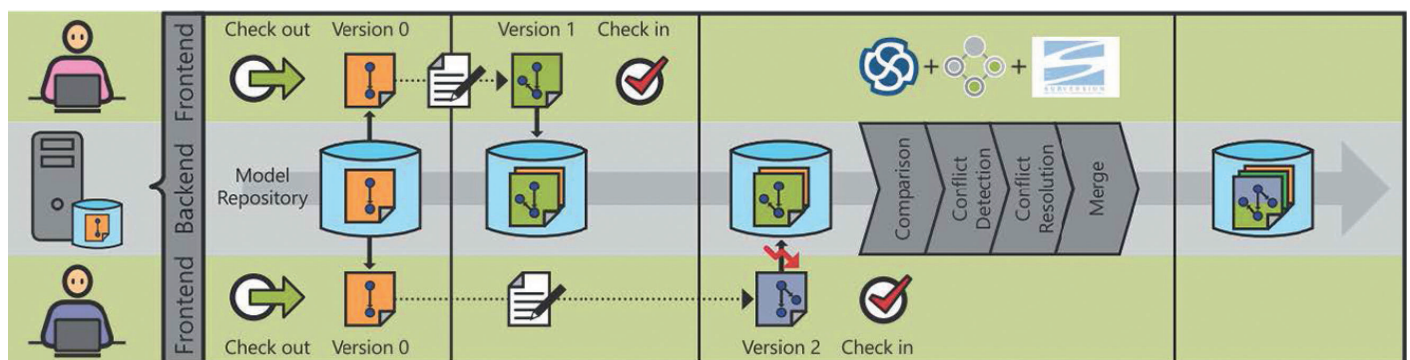


Neuer Ansatz für die Versionierung von Modellen

Neuer Ansatz für die Versionierung von Modellen

Wie können verteilte Teams effizient mit Softwaremodellen arbeiten? Der Beitrag stellt einen Lösungsansatz vor, der in Zusammenarbeit mit der Technischen Universität Wien erarbeitet wurde.

DR. OLIVER ALT, ROMAN BRETZ, DR. KONRAD WIELAND *



Überblick: Bei der optimistischen Versionierung wird dasselbe Artefakt zeitgleich bzw. parallel bearbeitet.

Für die modellbasierte Entwicklung mechatronischer Systeme wird gerne die Systems Modeling Language (SysML) verwendet, mit der sich Struktur und Verhalten eines Systems beschreiben lassen. Zudem erlaubt es die SysML, die textuellen Anforderungen an ein System in das Modell zu integrieren, sodass sich diese mit anderen Modellelementen verknüpfen lassen.

Die Sprache UML (Unified Modeling Language) kommt zum Einsatz, wenn Softwaresysteme realisierungsnäher beschrieben werden sollen. Anfangs für die Beschreibung objektorientierter Software konzipiert, lässt sich die UML auch für Embedded-Software verwenden, da sie Struktur und Verhalten eines Programms beschreibt. Dies kann so weit gehen, dass komplett lauffähiger Code aus einem UML-Modell generiert wird.

Verteilte Bearbeitung von Modellen

Die Bedeutung von Modellen für die Entwicklung nimmt stetig zu. Damit ergeben sich aber auch Anforderungen an die Infrastruktur und das Konfigurationsmanage-

ment von Modellen, etwa im Hinblick auf verteiltes Arbeiten. Dies gilt umso mehr, als sich die Arbeitswelt immer stärker globalisiert und Entwicklungsstandorte oft geografisch weit voneinander entfernt liegen.

In der Vergangenheit gab es bei Modellierungswerkzeugen wie Enterprise Architect (Sparx Systems) diverse Ansätze für eine verteilte Zusammenarbeit. Eine Möglichkeit ist die Nutzung einer zentralen Datenbank, auf die alle Modellnutzer in Echtzeit zugreifen. Damit sind Änderungen sichtbar, sobald sie gemacht werden. Das Problem bei einer gemeinsamen Datenbank ist aber die Schwierigkeit, einzelne Elemente oder Strukturen des Modells zu versionieren und einzeln freizugeben. Ein Lösungsansatz war es, Teilstrukturen des Modells zusätzlich zur Speicherung in der Datenbank auch in Dateien zu sichern und mit Versionskontrollsystemen zu verwalten. Diese Versionierung funktioniert aber maximal auf Paketebene und nicht für einzelne Elemente. Pakete sind vergleichbar mit Ordnern auf einer Festplatte, nur enthalten sie in der Modelldatenbank nicht Dateien, sondern Modellelemente, Verbindungen und Diagramme.

Besser wäre also eine feingranulare Versionierung, die die parallele Arbeit an Modellteilen ermöglicht und im Konfliktfall die

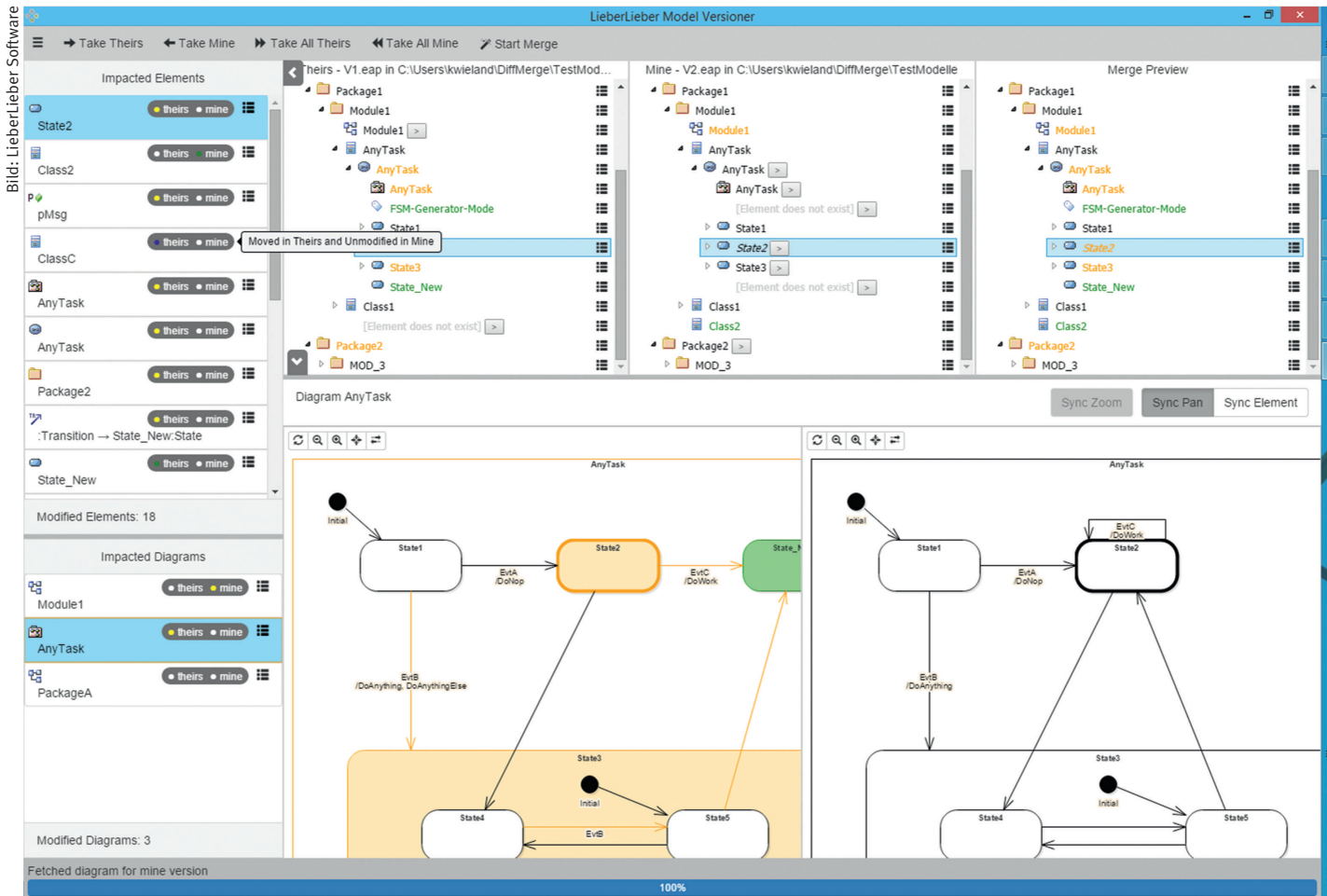
Änderungen strukturiert zusammenführt. Diese Konzepte kennt man schon einige Jahre von Konfigurationsmanagementsystemen für Quellcode. In Systemen wie Subversion oder Git werden Stände von Textdateien sogar teilweise parallel bearbeitet und später wieder werkzeuggestützt zusammengeführt.

Die dabei eingesetzten Prinzipien lassen sich grundsätzlich auch auf Modelle übertragen. Allerdings ist hier der Vergleich von Änderungen und die Zusammenführung anspruchsvoller als bei Textdaten, da hinter Modellen komplexere Datenstrukturen stecken. Da man es bei Modellen mit Diagrammen zu tun hat, muss auch definiert werden, ob etwa eine Verschiebung eines Elements im Diagramm schon als Änderung wahrgenommen wird oder nicht.

Basierend auf Forschungsergebnissen der Technischen Universität Wien zu Thema Modellversionierung konnte LieberLieber eine nutzerfreundliche State-of-the-Art-Lösung zum Verzweigen und Zusammenführen von Modellversionen für Enterprise Architect realisieren (LieberLieber Model Versioner).

Die Lösung erlaubt es, von einer Basisversion eines Modells zwei Versionen parallel weiter zu bearbeiten und die Änderungen nach Bearbeitung wieder zu einem konsolidierten Modell zusammenzuführen. Sollten

* Dr. Oliver Alt, Roman Bretz, Dr. Konrad Wieland
... arbeiten in leitenden Positionen bei LieberLieber Software in Wien.



Screenshot: Der LieberLieber Model Versioner vergleicht und visualisiert parallele Änderungen.

dabei keine Konflikte auftreten, die manuell aufzulösen sind, kann diese Konsolidierung automatisch im Hintergrund geschehen.

Das Verfahren basiert auf dem Konzept des Drei-Wege-Vergleichs, das bereits bei der Versionierung von Quelltext Anwendung findet. Dabei werden aus einer Basisversion zwei Versionen abgespalten und parallel bearbeitet. Durch den Vergleich des Basismodells mit den beiden abgezweigten Versionen lassen sich die Inhalte sehr gut wieder zu einer konsolidierten Version zusammenführen.

Feingranularer Drei-Wege-Vergleich von Modellversionen

Beim Drei-Wege-Vergleich von Modellen werden Modellelemente und Verbindungen sowie deren Attribute in den Vergleichsprozess mit einbezogen. Dadurch wird die feingranulare Nachverfolgung von Änderungen in einem bisher bei Modellierungswerkzeugen nicht erreichten Ausmaß ermöglicht.

Darüber hinaus lässt sich diese Methode der Versionierung mit bestehenden Konfigurationsmanagementsystemen koppeln. So

entsteht eine praktisch automatische, integrative Arbeitsweise im Umgang mit Modellversionen.

Zur Veranschaulichung ein Beispiel aus der Praxis: Zwei Entwickler holen sich den aktuellen Stand des Modells aus dem Konfigurationsmanagement und beginnen daran zu arbeiten. Beim Zurückspielen einer neuen Version startet automatisch die Modellzusammenführung. Gibt es keine Konflikte, so läuft der Vorgang im Hintergrund ab. Im Fall von Konflikten erhält der Benutzer eine Meldung und kann sie manuell auflösen. Dann werden die Änderungen zusammengeführt. Es entsteht eine konsolidierte, neue Version.

Der Screenshot zeigt die Benutzeroberfläche des Model Versioner. Die veränderten Teile der Modelle werden im oberen Teil als Baumstruktur dargestellt. Man sieht die beiden Versionen und eine Vorschau auf das Modell, das nach dem Zusammenführen entsteht. Der Benutzer kann bei Bedarf in die automatische Zusammenführung eingreifen oder einen Konflikt lösen. Im unteren Teil lässt sich eine Diagrammvorschau einblen-

den, die grafisch zeigt, was wie geändert wurde. Bei der vollwertigen Integration in ein Konfigurationsmanagement erscheint das Fenster nur, wenn es durch den Benutzer manuell zu lösende Aufgaben gibt, die nicht automatisch konsolidiert werden können.

Fazit: Die verteilte Bearbeitung von Modellen ist möglich, wenn es Werkzeugunterstützung für das Verzweigen und Zusammenführen von Modellen gibt. Mit der vorgestellten Lösung werden diese Aufgaben auf feingranularer Ebene erledigt. Dabei kann der Anwender Regeln für die automatische Zusammenführung definieren, die auf Prozesse eines Kunden abgestimmt werden. Selbst die Anzahl der Details, die dem Benutzer angezeigt werden, lässt sich kundenspezifisch einstellen. Damit steht für die modellbasierte Entwicklung ein Werkzeug zur Verfügung, das ein lange eher schlecht gelöstes Infrastrukturproblem adressiert. // FG



ENTERPRISE ARCHITECT

Enterprise Architect 12.1 mit UML 2.5 Unterstützung

Softwareentwicklung kann so einfach werden:

- ⇒ Unterstützung für UML, SysML, SOMF, BPMN, BPEL uvm.
- ⇒ Besseres Design & Verständnis der Architektur im gesamten Team
- ⇒ Validieren der Anforderungen
- ⇒ Nachvollziehbarkeit der Testfälle und der Testabdeckung
- ⇒ Unterstützung verschiedener Vorgehensmodelle
- ⇒ Für mehr als 10 Programmiersprachen



Jetzt Enterprise Architect 12.1 testen!
Besuchen Sie www.sparxsystems.de

LieberLieber 

UML Debugging

Embedded Code Generation

Requirements Tracing

Model Versioning

Addins & Customizations

Make Enterprise
Architect your tool.