



# Synchronisierungsstrategien

Seminar Modellgetriebene Softwareentwicklung  
Zwischenpräsentation

Stefan Mertins

# Inhalt

---

## ■ Einleitung

## ■ Synchronisation

- ▶ Szenarien
- ▶ Strategien
- ▶ Differenzbildungsansatz
- ▶ Eigenschaften
- ▶ Konflikte

## ■ Anwendungsfall

- ▶ Visio-EMF Bridge
- ▶ EMF Compare
- ▶ Lösungsansatz



# Einleitung

---

## ■ Modelle spielen in der Informatik eine wichtige Rolle

- ▶ Große Artenvielfalt an Modellen (für viele Anwendungsformen existieren verschiedene Arten von Sichtweisen → Modellen)

## ■ Modelle unterliegen vielen Änderungen

- ▶ Modelle die bspw. eine Sicht auf eine Anwendung oder Prozesse darstellen, unterliegen gewöhnlich genauso viele Änderungen wie diese selbst

## ■ Viele verschiedene Werkzeuge auf dem Markt

- ▶ Werkzeuge bieten unterschiedliche Vor- und Nachteile
- ▶ Werkzeuge in verschiedenen Werkzeugräumen
- ▶ Werkzeuge nutzen unterschiedliche Formate

## ■ Modelle haben verschiedene

- ▶ Repräsentationen
- ▶ Versionen



# Synchronisation

---

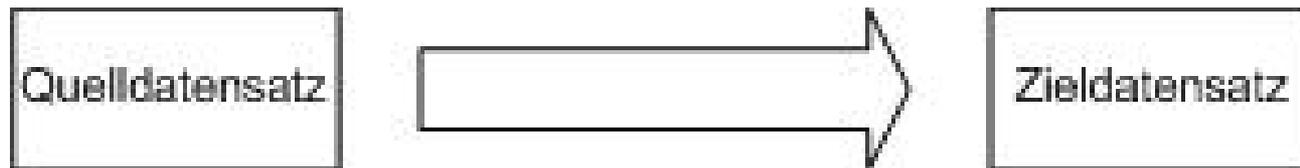
- **Synchronisation kommt aus dem Griechischen von *sýn* für "zusammen" und *chrónos* für "Zeit"**
- **bedeutet das "Herstellen von Gleichzeitigkeit"**
  - ▶ In unserem Fall geht es jedoch um die Herstellung von Gleichheit
- **Prozess, nachdem Datensätze einen äquivalenten Inhalt nach Abschluss haben**
  - ▶ Inhalt muss nach Abschluss nicht Identisch sein
- **Ansatz zur Konsistenzverwaltung**
  - ▶ Synchronisation ist Auflösung von Inkonsistenzen zwischen Datensätzen
- **Synchronisation kann auch auf Modelle angewandt werden**
  - ▶ Kann bei Beherrschung der Modellvielfalt eingesetzt werden



# Szenarien

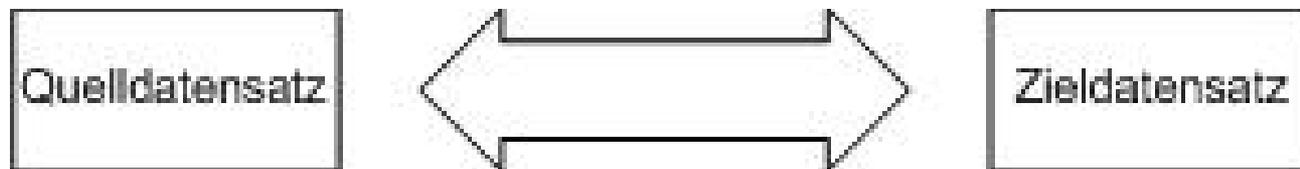
## ■ 1-Wege

- ▶ Unidirektionalen Synchronisationen
- ▶ Übertragung von Daten von Quelldatensatz zu Zieldatensatz
- ▶ Vorhandene Daten im Zieldatensatz werden überschrieben
- ▶ Bsp.: Abruf von Emails mittels Imap Protokoll



## ■ 2-Wege

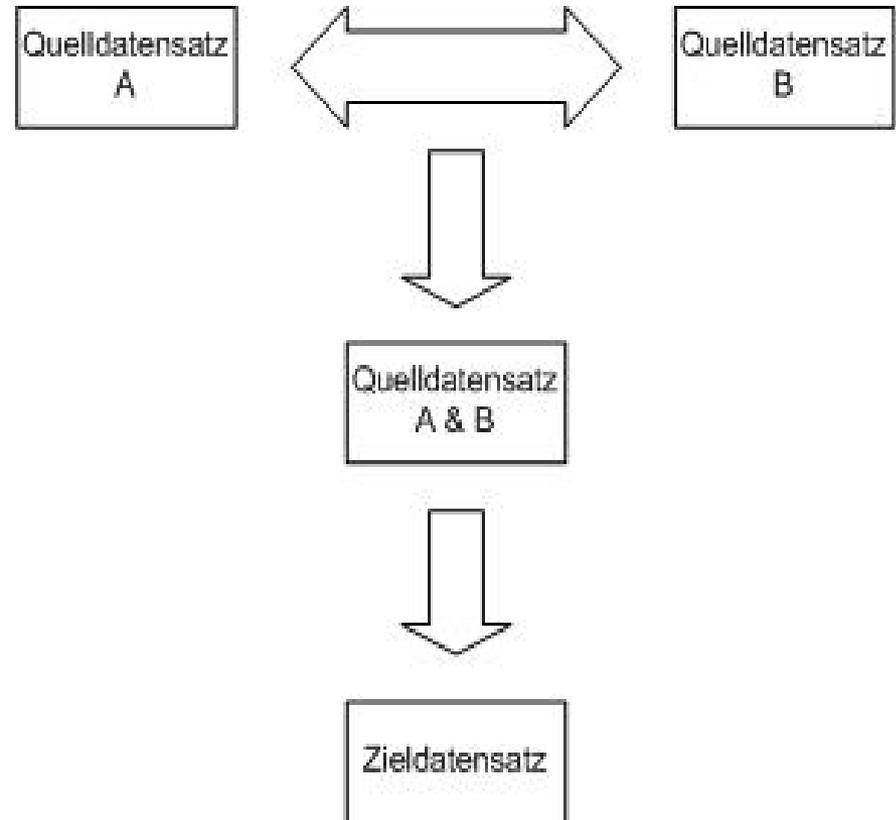
- ▶ Bidirektionale Synchronisation
- ▶ Übertragung von Daten zwischen Quelldatensatz und Zieldatensatz
- ▶ Häufig auftretende Form der Synchronisation
- ▶ Bsp.: Abgleich von Terminen zwischen PC und PDA



# Szenarien

## ■ 3-Wege

- ▶ Bidirektionale Synchronisation
- ▶ Übertragung von Daten zwischen zwei Quelldatensätzen und einem Zieldatensatz
- ▶ Synchronisation der beiden Quelldatensätze
- ▶ Anschließend Synchronisation mit Zieldatensatz
- ▶ Bsp.:  
Versionsverwaltungssystem auf dem zwei Personen gleichzeitig eine geänderte Datei commiten



# Strategien

---

## ■ Protokoll basierter Ansatz

- ▶ Änderung an Quelldatensatz werden protokolliert
- ▶ Resultierendes Protokoll mit 100% der Änderungen ermöglicht die Nachvollziehbarkeit und ein wiederholtes Ausführen auf Zieldatensatz
- ▶ Probleme beim verteilten arbeiten (3-Wege Synchronisation)
- ▶ Erfordert lückenlose Protokolle

## ■ Differenzbildungsansatz

- ▶ Datensätze werden miteinander verglichen und eine Menge der Unterschiede wird erstellt (Differenz)
- ▶ Differenz kann anschließend genutzt werden um Unterschiede aufzulösen
- ▶ Ansatz Rechenintensiv, da Datensätze verglichen und eine Differenz erstellt werden muss
- ▶ Wird in drei bis vier Phasen unterteilt (Matching/Mapping, Differenzbildung, Merging)



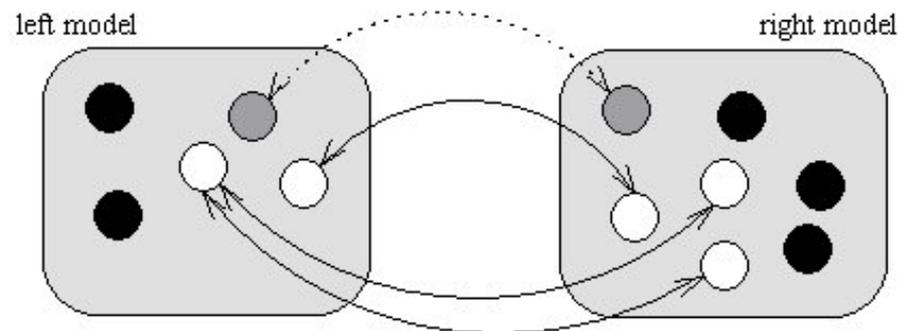
# Differenzbildungsansatz

## ■ Matching

- ▶ Zuordnung von Elementen ermöglichen
- ▶ UUID (Universal Unique Identifiers)
- ▶ Matchingalgorithmus

## ■ Mapping

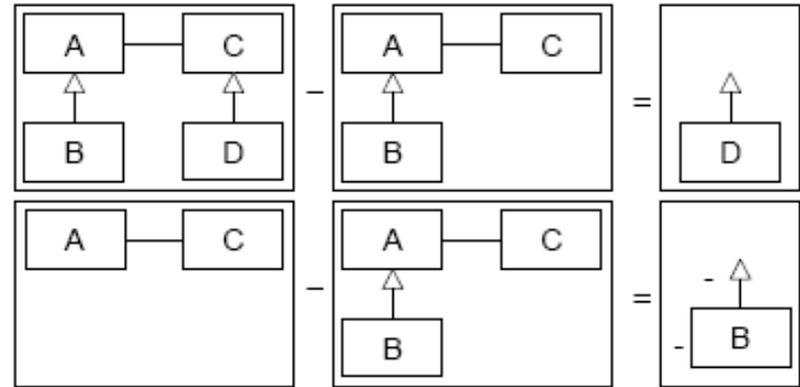
- ▶ Finden von zusammengehörenden Elementen auf Basis des Matchings
- ▶ Elemente mit konformen Elementen
- ▶ Elemente mit nicht konformen Elementen
- ▶ Elemente ohne Übereinstimmung



# Differenzbildungsansatz

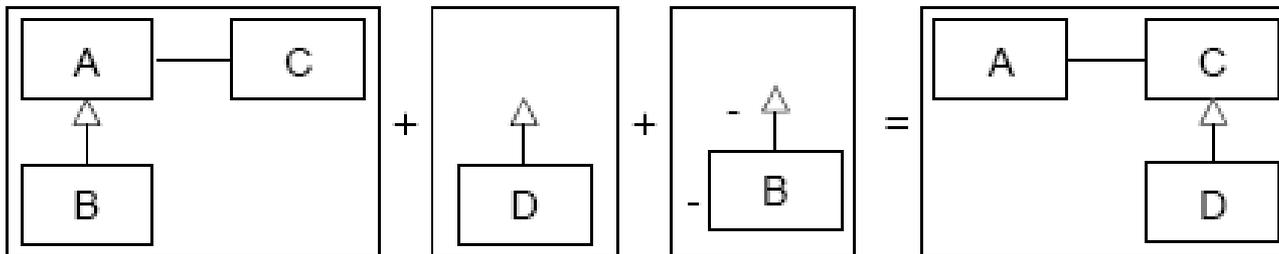
## ■ Differenzbildung

- ▶ Nähere Betrachtung von Elementen, die keine konforme, bzw. keine Übereinstimmung haben
- ▶ Identifizieren der Änderungen (editieren, hinzufügen, löschen)



## ■ Merging

- ▶ Zusammenführen der Differenz und des Zieldatensatzes



# Eigenschaften

---

## ■ Stabilität

- ▶ Wenn weder Quell- noch Zieldatensatz eine Änderung enthalten, darf der Synchronisationsprozess auch keine Änderung durchführen.

## ■ Bewahrung

- ▶ Änderungen, die vor Beginn des Synchronisationsprozesses vorhanden waren, müssen auch danach vorhanden sein.

## ■ Ausbreitung

- ▶ Änderungen, die vor Beginn des Synchronisationsprozesses vorhanden waren, müssen danach in allen Datensätzen vorhanden sein.

## ■ Zusammensetzbarkeit

- ▶ Egal in welcher Reihenfolge Synchronisationen hintereinander ausgeführt werden, das Ergebnis muss immer das gleiche sein.



# Konflikte

---

- **Probleme die der Synchronisierungsprozess eigenständig nicht lösen kann, wenn er die Eigenschaften einhält**
- **Einige Konflikte können durch die Anwendung von geschickten Algorithmen umgangen werden**
- **Meist erfordert das Auftreten eines Konflikts jedoch die Interaktion eines Benutzers**
  
- **Abstammung von Änderungen**
  - ▶ Unklar welches Modell die Änderungen enthält
  - ▶ Wurde ein Element gelöscht oder hinzugefügt?
  - ▶ Umgehen, durch gerichtete Synchronisation
- **Parallele Änderung eines Elements**
  - ▶ Bei zwei Änderungen am gleichen Element zur gleichen Zeit
  - ▶ Welche Änderung ist zu übernehmen?
  - ▶ Kann nur durch Nutzereingabe gelöst werden

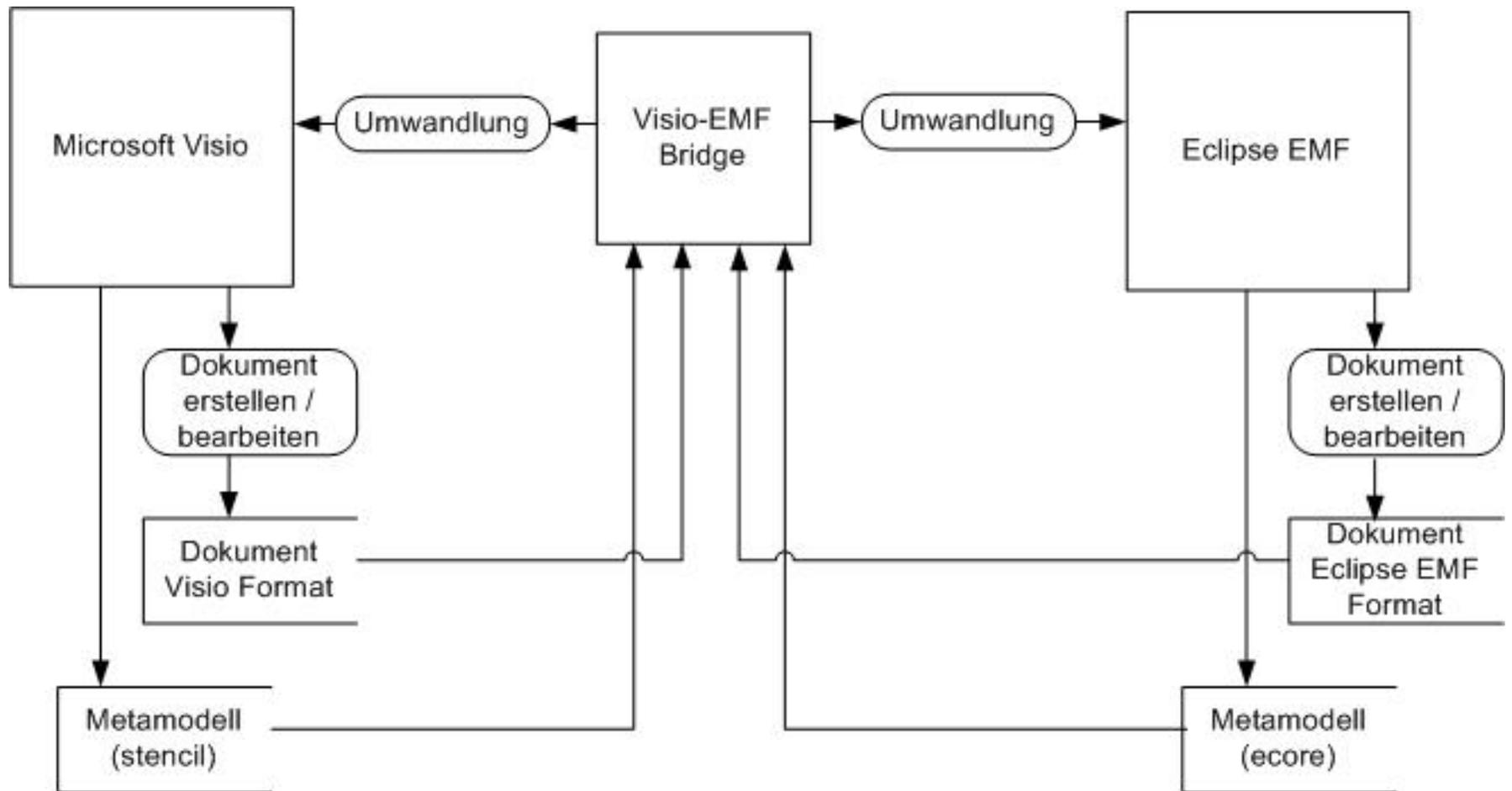
# Anwendungsfall

---

- **Arbeit liegt ein konkreter Anwendungsfall zu Grunde**
  - ▶ Visio-EMF Bridge
- **Eingehende Problematik von Modellversionsvielfalt soll mithilfe von Modellsynchronisation gelöst werden**
  - ▶ Theoretische Grundlage nun vorhanden
  - ▶ Praktische Umsetzung gewünscht
- **Einsatz bereits vorhandener Werkzeuge**
  - ▶ EMF Compare
  - ▶ Eigene Entwicklung eines Synchronisationsansatzes unnötig und unrealistisch



# Visio-EMF Bridge



## EMF Compare

---

- **EMF Compare ist Bestandteil des EMF Technologie Projekts**
- **Bietet Möglichkeit der Modellvergleiche**
- **Generische Unterstützung von Metamodellen**
- **Im Eclipse Modeling Bundle enthalten und hat deshalb eine weite Verbreitung**
- **Bereitstellung eine EMF Model Differenzengine, die Modelle mit dem gleichem Metamodell vergleicht und ein Model als Ausgabe liefert**
- **Unterstützung von 2 oder 3 Modellen als Eingabe**
- **Generischen und effizienten Vergleichsalgorithmus für Modelle**
- **Gleicher Werkzeugraum wie Visio-EMF Bridge**
- **Einfacher programmatischer Aufruf mit wenig Aufwand**

## ■ Realisierung in Form eines Eclipse Plugins

- ▶ Gleicher Werkzeugraum wie Visio-EMF Bridge und EMF Compare

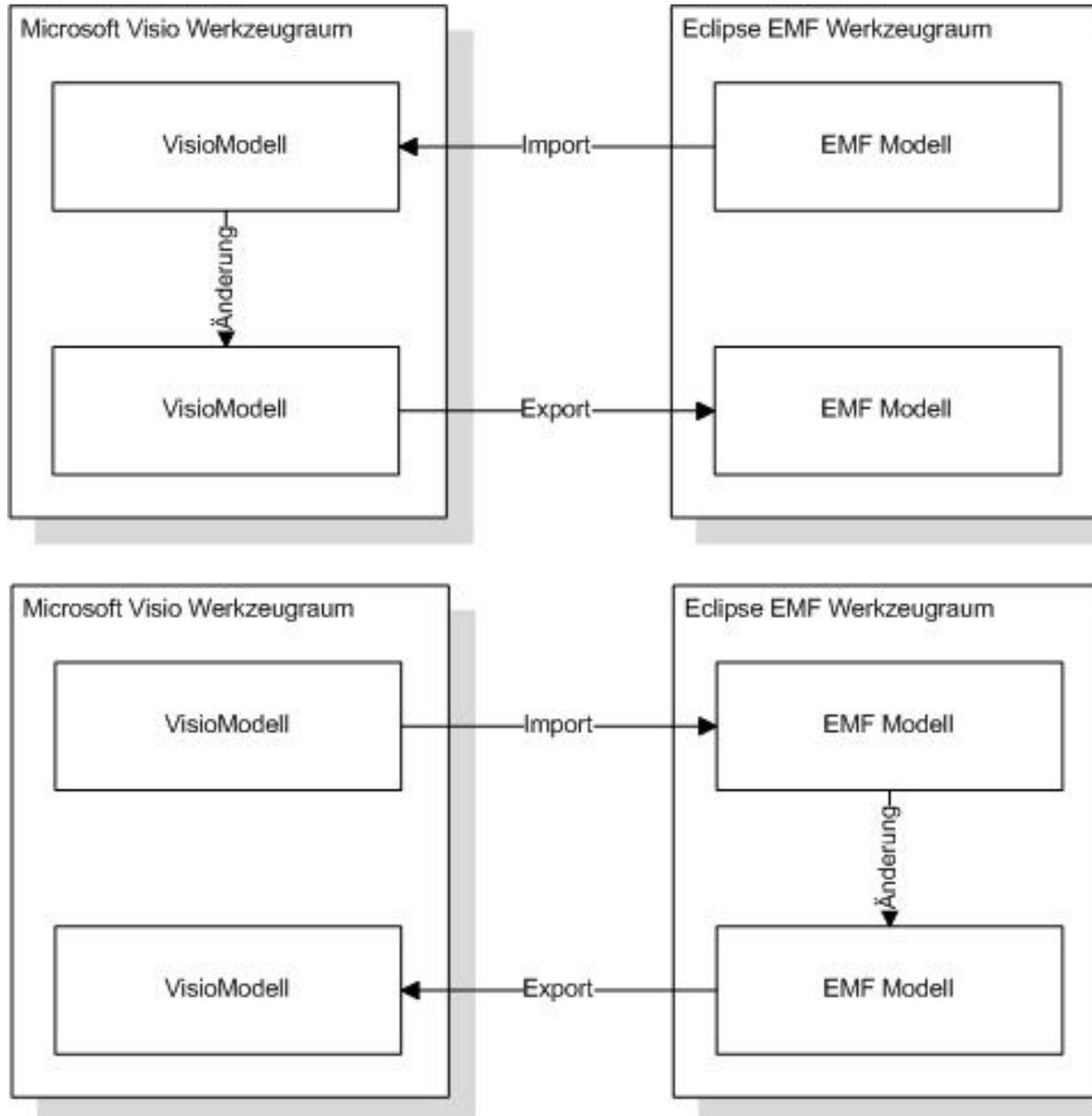
## ■ Einsatz von EMF Compare als Differenzengine

### ■ Aufgaben:

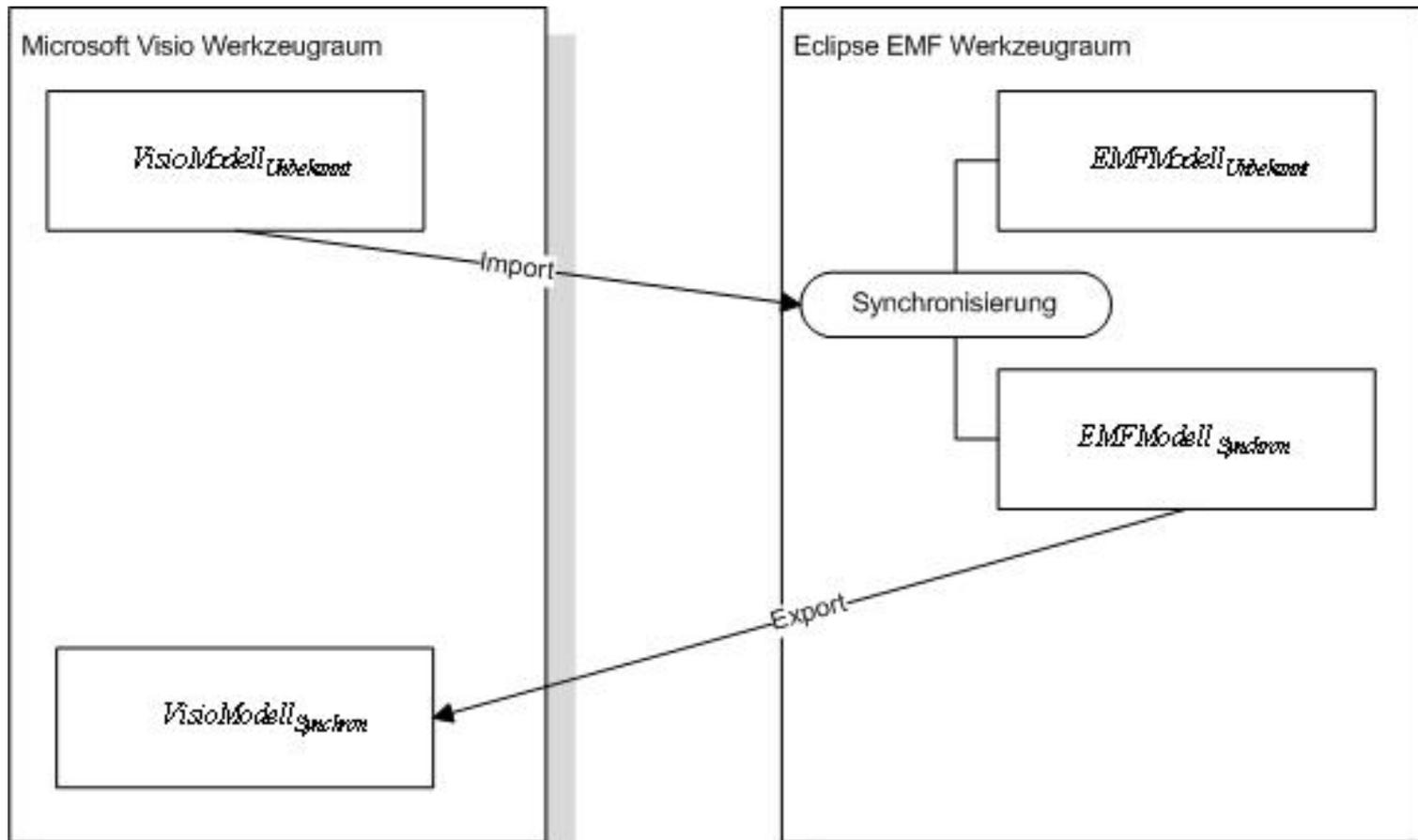
- ▶ Verwalten und speichern von zu synchronisierenden Modellen und dazugehörigen Metamodellen
- ▶ Erkennen von Änderungen der überwachten Modelle
- ▶ Überführen der Modelle in den jeweiligen Werkzeugraum mittels Visio-EMF Bridge
- ▶ Aufruf der Differenzengine für Modelle, die einer Änderung unterlagen

## ■ Drei konkrete Anwendungsfälle können auftreten

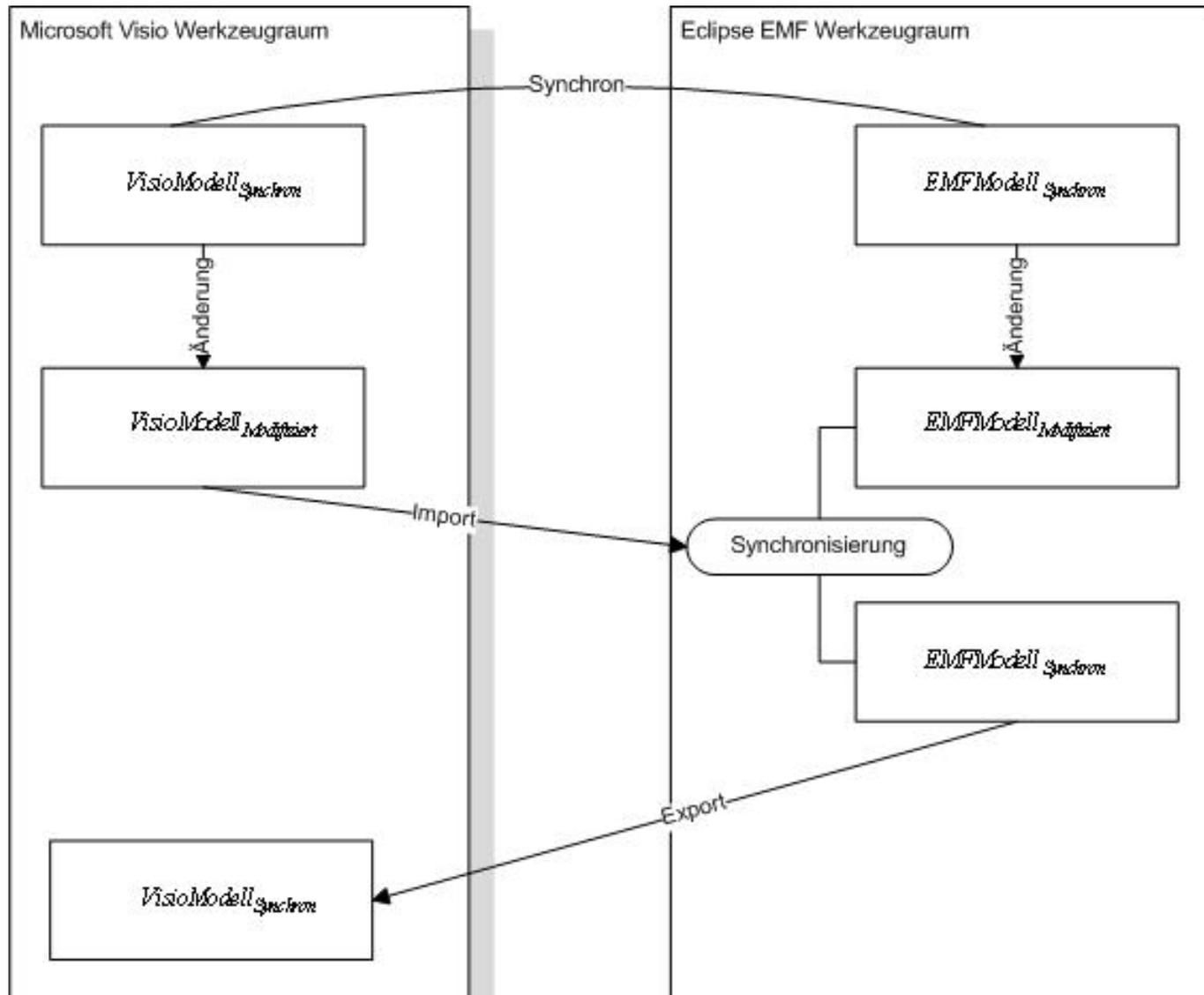
# Lösungsansatz



# Lösungsansatz



# Lösungsansatz





**Fragen?**