Diplomarbeitspräsentationen der Fakultät für Informatik

Konflikterkennung in der Modellversionierung

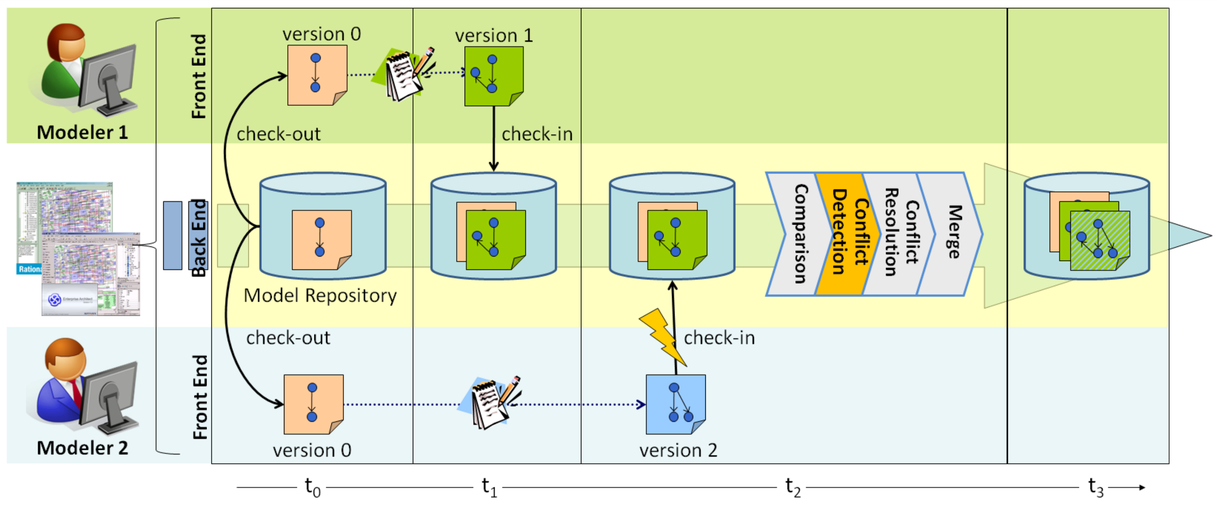
Philip Langer

Technische Universität Wien
Institut für Softwaretechnik und Interaktive Systeme
Arbeitsbereich: Business Informatics Group
Betreuerin: O.Univ.-Prof. Mag. Dipl.-Ing. Dr.techn. Gerti Kappel

FAKULTÄT FÜR !NFORMATIK

Master-/Diplomstudium:
Wirtschaftsinformatik

Kontext der Konflikterkennung



Komponenten der Konflikterkennung

Anforderungen an die Konflikterkennung

Zuverlässigkeit und Genauigkeit	Adaptierbarkeit
Sprachunabhängigkeit	Editorunabhängigkeit

Ursprungsmodell

Rechnungsdrucker	Mitarbeiter	Kunde
+ Rechnungsdrucker() + drucke(Kunde k)	+ Mitarbeiter() + getName1() + getName2()	+ Kunde() + getName1() + getName2()

Sprachunabhängiger Match

Der Match zwischen den Ursprungselementen und den Elementen der Arbeitskopie wird direkt nach dem *Checkout* durchgeführt, wenn das Ursprungsmodell und die Arbeitskopie identisch sind. Es werden IDs vergeben, um Elemente auch nach starker Veränderung wieder zu erkennen.

Sprachunabhängige Differenzermittlung

Auf Basis des zuvor erstellten Matchs wird nun ein generischer, statusbasierter 2-Wegsvergleich zwischen dem Ursprungsmodell und dem jeweils veränderten Modell (a) und (b) durchgeführt und so die atomaren Operationen ermittelt. Das Ergebnis sind zwei Differenzmodelle.

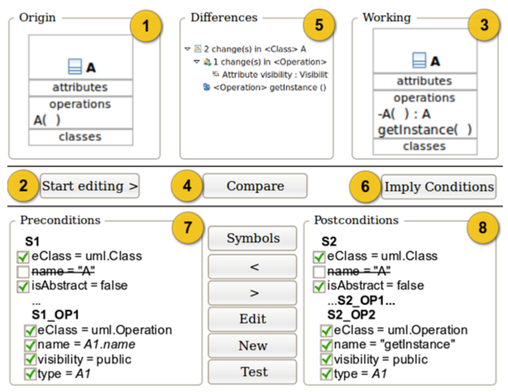
Erkennung zusammengesetzter Operationen und Refactorings

Definition

Zusammengesetzte Operationen und Refactorings sind metamodellabhängig und müssen daher vom Benutzer definiert werden. Hierfür wird eine anwenderfreundliche, beispielgetriebene Vorgehensweise verwendet. Der User erstellt im Bereich *Origin* (1) das Ausgangsmodell und führt im Bereich *Working* (3) die Änderung der zu definierenden Operation durch. Die Operationen (5) werden durch einen 2-Wegsvergleich ermittelt und die Vor- (7) und Nachbedingungen (8) der zusammengesetzten Operation automatisch abgeleitet. Diese Bedingungen können vom User anschließend noch weiter verfeinert werden.

Erkennung

Zur Erkennung wird im atomaren Differenzmodell nach dem Differenzmuster (5) gesucht und die Vorbedingungen (7) und Nachbedingungen (8) geprüft.



Convert to Singleton-Refactoring Definition

Konflikterkennung

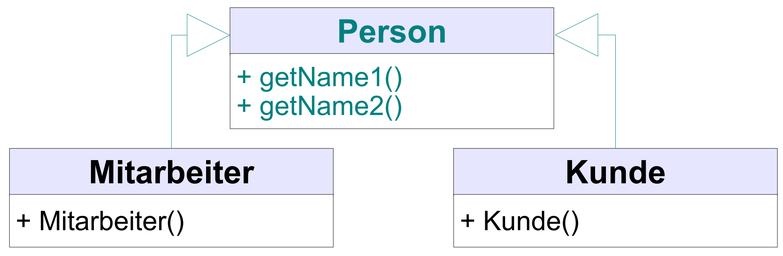
Die Konflikte werden einerseits über die generische und sprachspezifische Analyse der beiden Differenzmodelle (Ursprung zu (a) bzw. (b)) ermittelt und andererseits über die Simulation der Zusammenführung und Validierung dieser Zusammenführung nach sprachspezifischen Regeln.

Verändertes Modell (a)

Rechnungsdrucker	Mitarbeiter	Kunde
- Rechnungsdrucker() + getInstance() + drucken(Kunde k)	+ Mitarbeiter() + getVorname() + getNachname()	+ Kunde() + getVorname() + getNachname()

Verändertes Modell (b)

RDruckservice
+ RDruckservice() + druckeKunde(Kunde k)



Konfliktbericht ohne Refactoring-Erkennung

- **Mitarbeiter**
  - *DeleteUpdate*-Konflikt: `getVorname()`
  - *DeleteUpdate*-Konflikt: `getNachname()`
- **Kunde**
  - *DeleteUpdate*-Konflikt: `getVorname()`
  - *DeleteUpdate*-Konflikt: `getNachname()`
- **Rechnungsdrucker**
  - *ChangeChange*-Konflikt: `drucken()` <> `druckeKunde()`

Konfliktbericht mit Refactoring-Erkennung

- **Rechnungsdrucker**
  - *ChangeChange*-Konflikt: `drucken()` <> `druckeKunde()`

Petra Brosch, Philip Langer, Martina Seidl, and Manuel Wimmer. Towards End-User Adaptable Model Versioning: The By-Example Operation Recorder
Accepted for the *International Workshop on Comparison and Versioning of Software Models (CVSM09) colocated with Int. Conf. on Soft. Eng. (May) 2009.*

Kontakt: langer@big.tuwien.ac.at