

Term Selection Tool

- TeSeT -

Abschlussvortrag

Anna Björk Nikulásdóttir

Ablauf des Vortrags

- Einleitung
 - Aufgabenbeschreibung, Terminologie
- Kurze Demo von TeSeT
- Datenstrukturen
- Programmablauf
- TeSeT Demo
- Kritische Beurteilung vom Vorgehen

Einleitung

- Das Projekt **Term Selection Tool (TeSeT)** wurde am Fraunhofer IPSI in Darmstadt durchgeführt.
- Auftraggeber ist das Bibliographische Institut & F. A. Brockhaus AG in Mannheim.
- Dort wurde ein Werkzeug gebraucht, das lexikographische Arbeit an Komposita unterstützt.

Zielgruppe

- Die Zielgruppe von TeSeT umfasst in erster Linie studentische Hilfskräfte mit Linguistikkenntnissen, außerdem werden Redakteure das Werkzeug benutzen.
- Es konnte nicht von überdurchschnittlichen Computerkenntnissen ausgegangen werden.
- Wichtig war die Berücksichtigung der „Massenarbeit“: der Benutzer muss in der Regel mehrere Tausend Komposita bearbeiten.

Programmiersprache

- Programmiert wurde in Smalltalk
- Smalltalk ist eine rein objektorientierte Sprache
- Entwicklung in der Entwicklungsumgebung Visual Works
- Der Quellcode der gesamten Smalltalk Klassenbibliothek steht zur Verfügung und kann beliebig ergänzt und verändert werden

Aufgabenbeschreibung

Termzuordnung bei Komposita

- Bei deutschen Komposita sollen den beiden Hauptkomponenten die richtigen Definitionen zugeordnet werden.
- Die ersten Dateien wurden manuell bearbeitet. Der Bearbeiter musste die notwendigen Informationen aus verschiedenen Quellen zusammen suchen und seine Ergebnisse in Tabellen eintragen.
- Bei ca. 120.000 Komposita ein enormer Aufwand!

Aufgaben von TeSeT

- **Kernfunktionen**

- Alle benötigten Informationen in einem Fenster verfügbar machen
- Das Ausfüllen der Tabellen übernehmen

- **Usability**

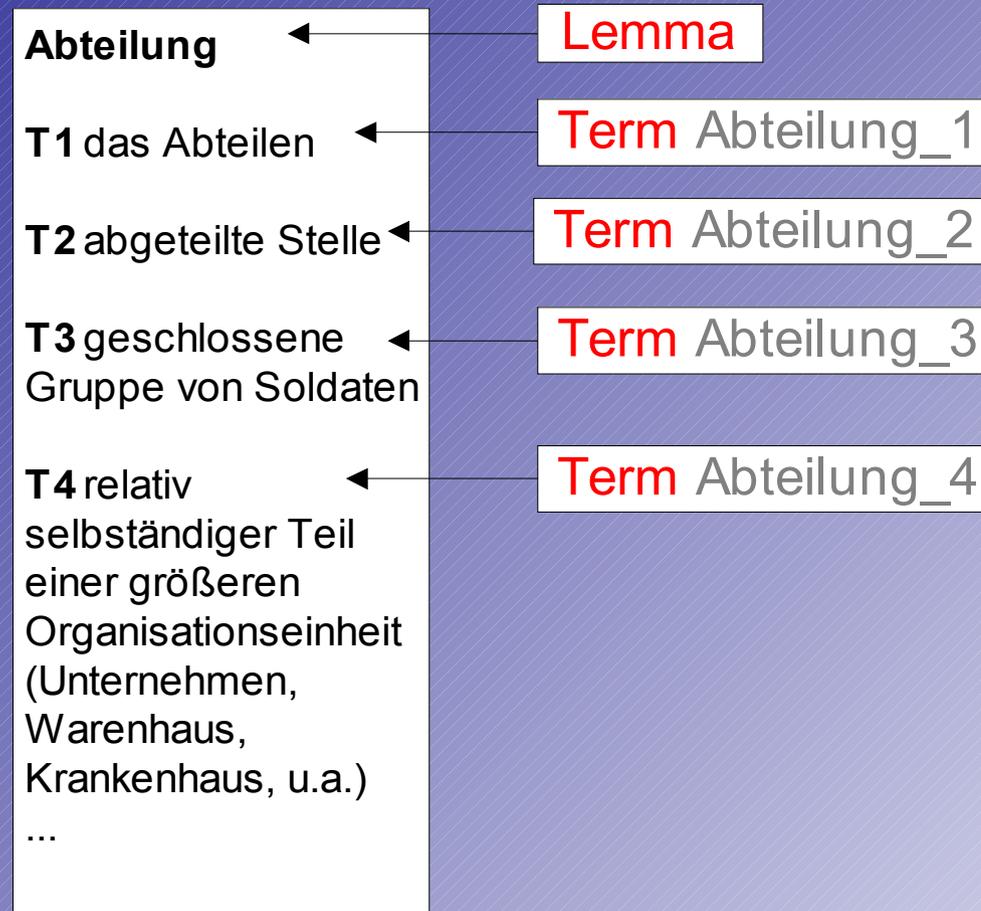
- Verschiedene Funktionen anbieten, die der Übersicht über die Arbeit dienen
- Die GUI so gestalten, dass für jeden Arbeitsschritt möglichst wenig geklickt werden muss

Terminologie - Lemmata und Terme

- Ein **Lemma** ist ein Stichwort im Wörterbuch
- Unter jedem **Lemma** stehen eine oder mehrere Definitionen .
- Jede dieser Definitionen ist einem **Term** zugeordnet. Die **Terme** werden nicht explizit aufgeführt.
- Ein **Lemma** hat nicht eine eindeutige Definition.
- Ein **Term** hat immer eine eindeutige Definition

Lemmata - Terme

Beispiel



Termzuordnung bei Komposita – Beispiel

Abteilungsleiter

Abteilung -s- Leiter

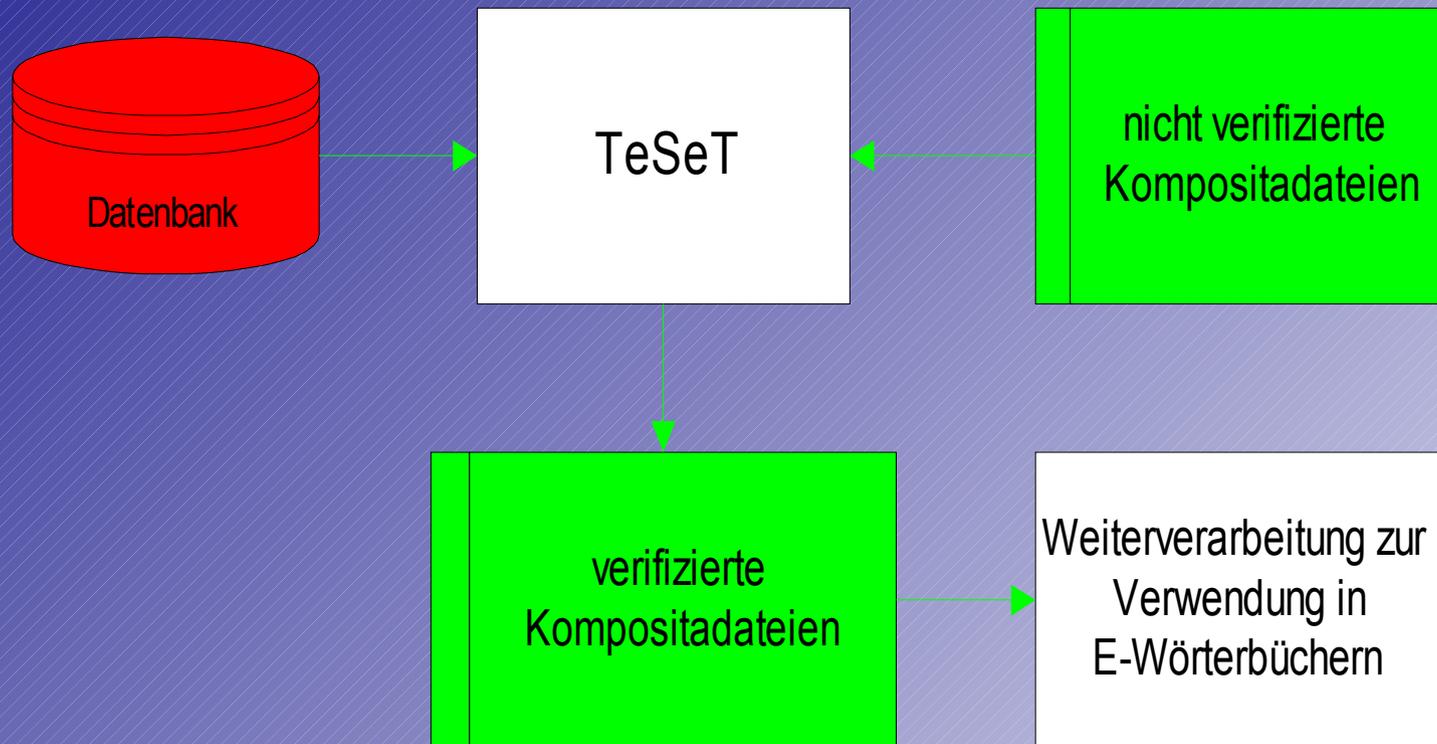
T1 das Abteilen
T2 abgeteilte Stelle
T3 geschlossene Gruppe von Soldaten
T4 relativ selbständiger Teil einer größeren Organisationseinheit (Unternehmen, Warenhaus, Krankenhaus, u.a.)
...

T1 Gerät mit Sprossen od. Stufen zum Hinauf- u. Hinuntersteigen
T2 jmd. der etw. leitet, der leitend an der Spitze von etw. steht
T3 Stoff, der Energie leitet

Benötigte Daten

- Die zu bearbeitenden Komposita
 - Diese sind in Tabellendateien organisiert
- Definitionen der einzelnen Terme
 - Diese sind in einer Datenbank vorhanden

Die Datensicht

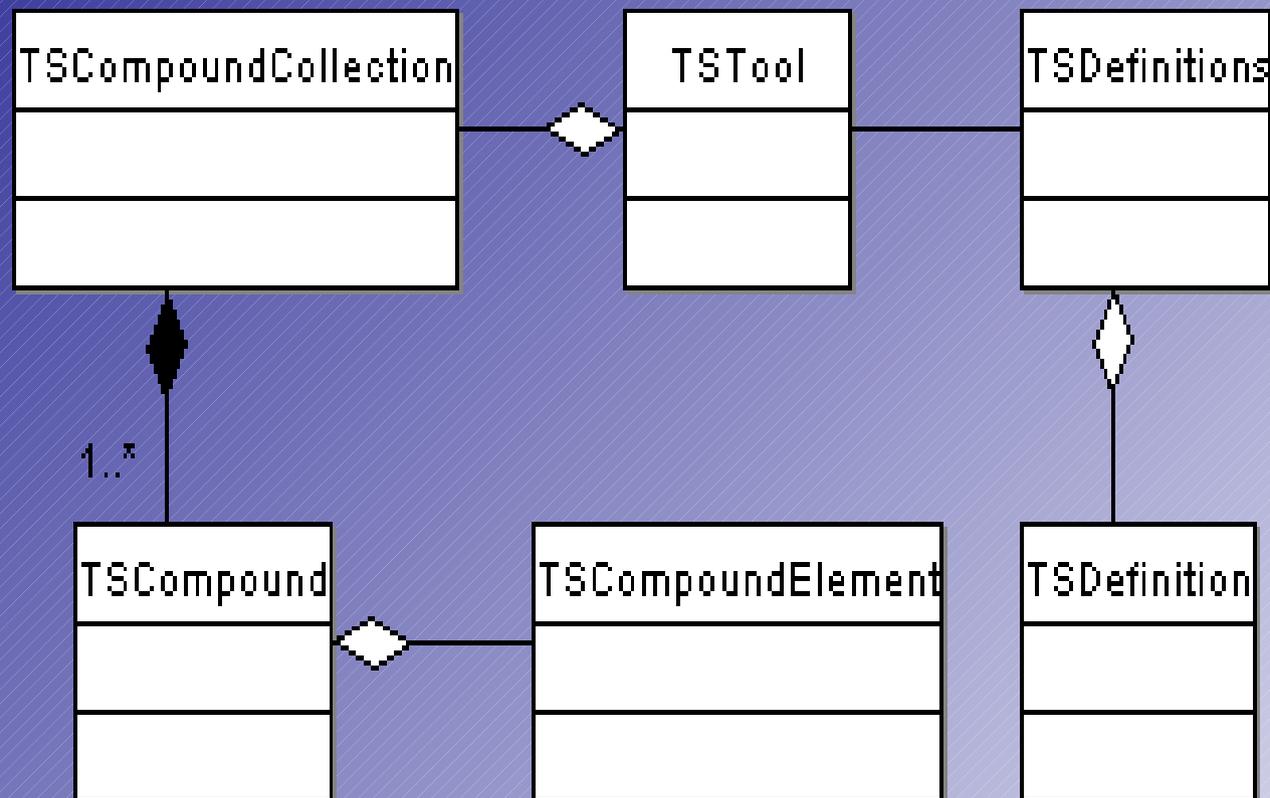


Vorhandene Klassen

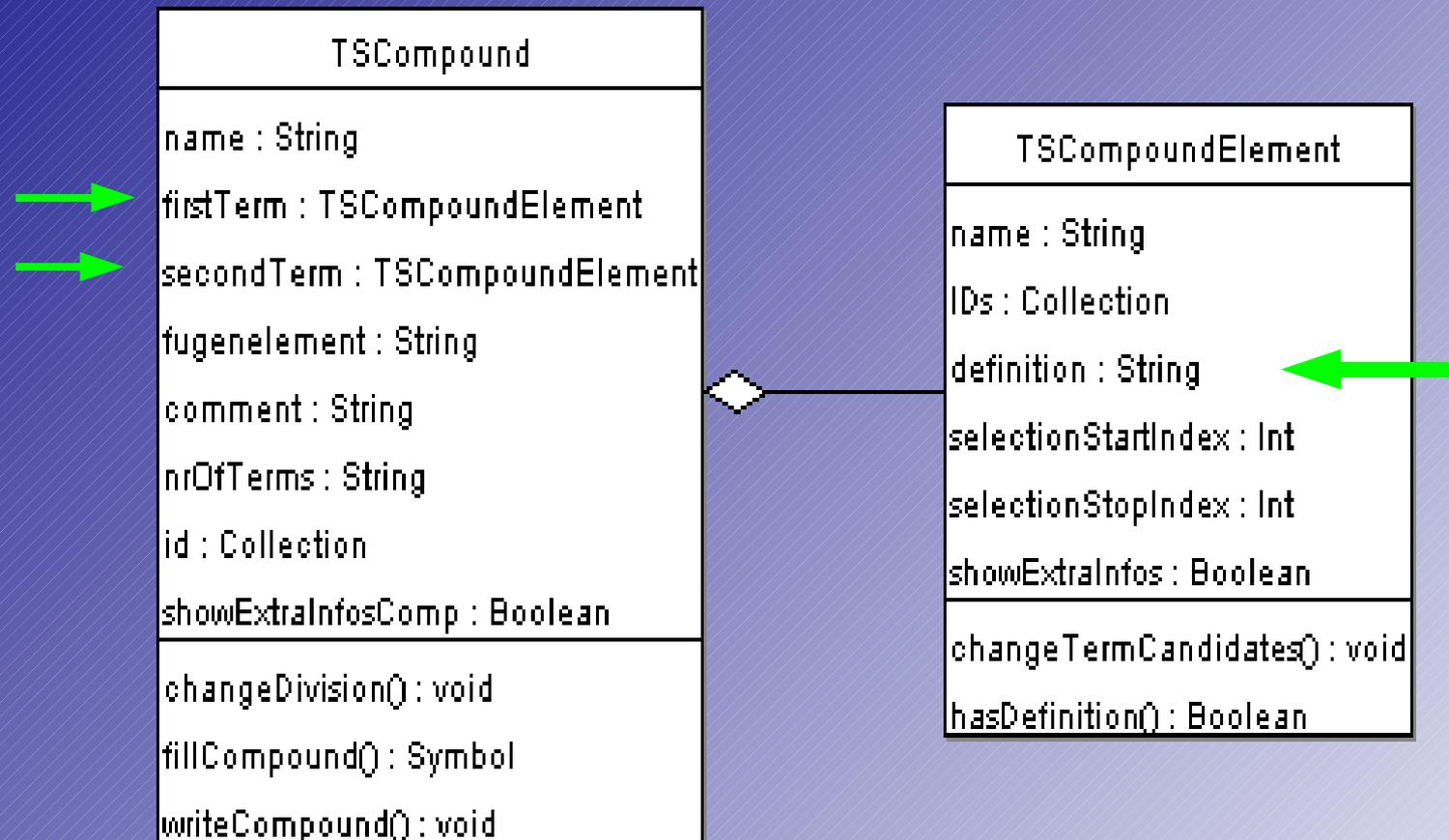
- Folgende Klassen stehen schon zur Verfügung:
 - Die Datenbank-Applikation und zugehörige Klassen
 - Klassen, welche die Texte aus der Datenbank parsen.
- Es müssen geringfügige Anpassungen dieser Klassen für TeSeT gemacht werden.

Datenstrukturen

Klassendiagramm



Kompositaklassen



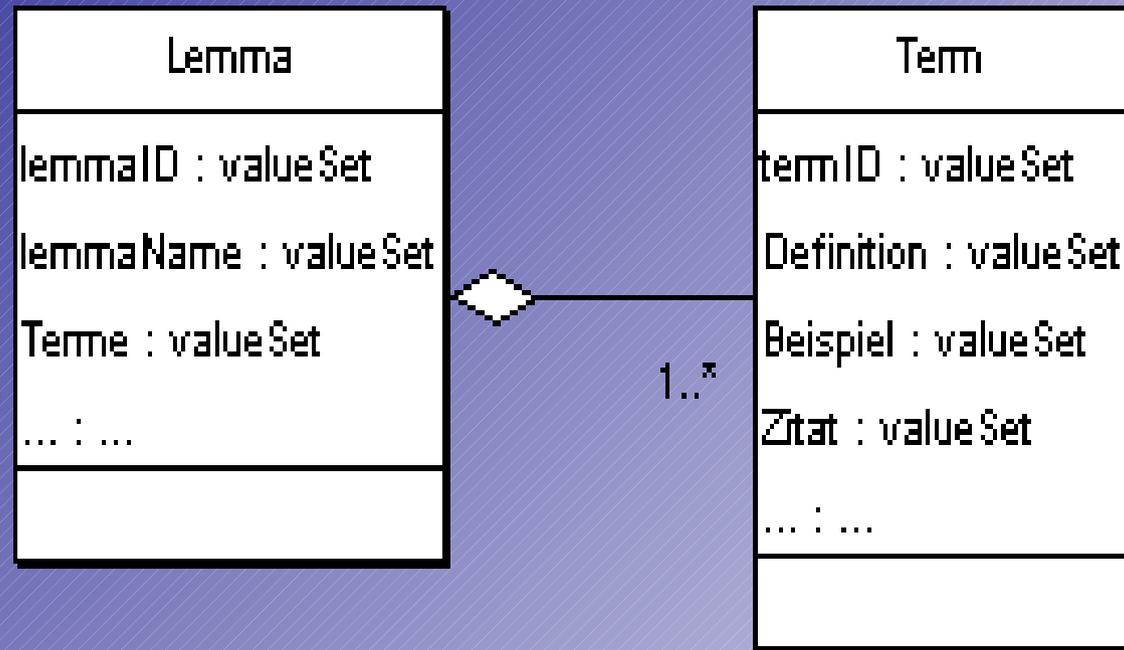
Kompositacontainer

TSCollection
collectionFromTabFile() : TSCollection filter() : TSCollection

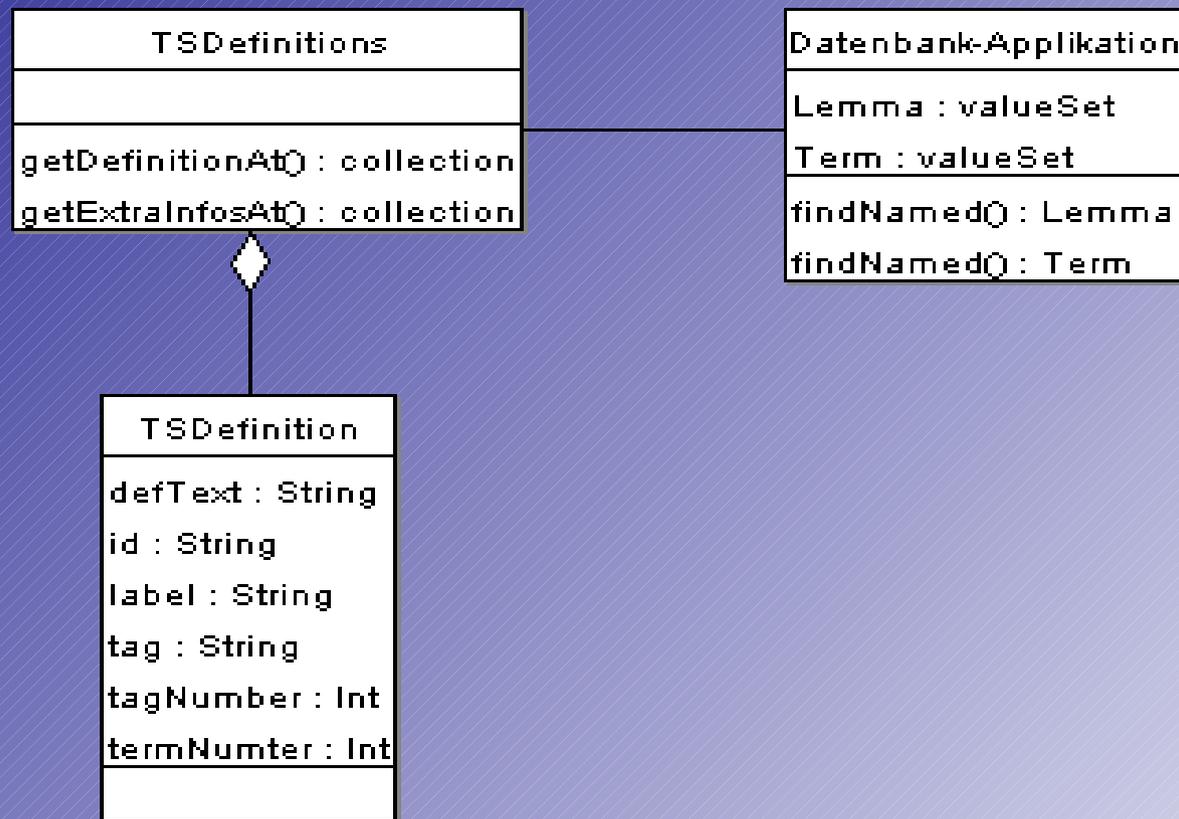
Applikationsklasse

TSTool
<code>compounds : TSCollection</code>
<code>changedCompoundSelection() : void</code>
<code>changedSelectionFrom() : void</code>
<code>updateCompoundsList() : void</code>
<code>setTextEditorValue() : text</code>

Datenbankschema

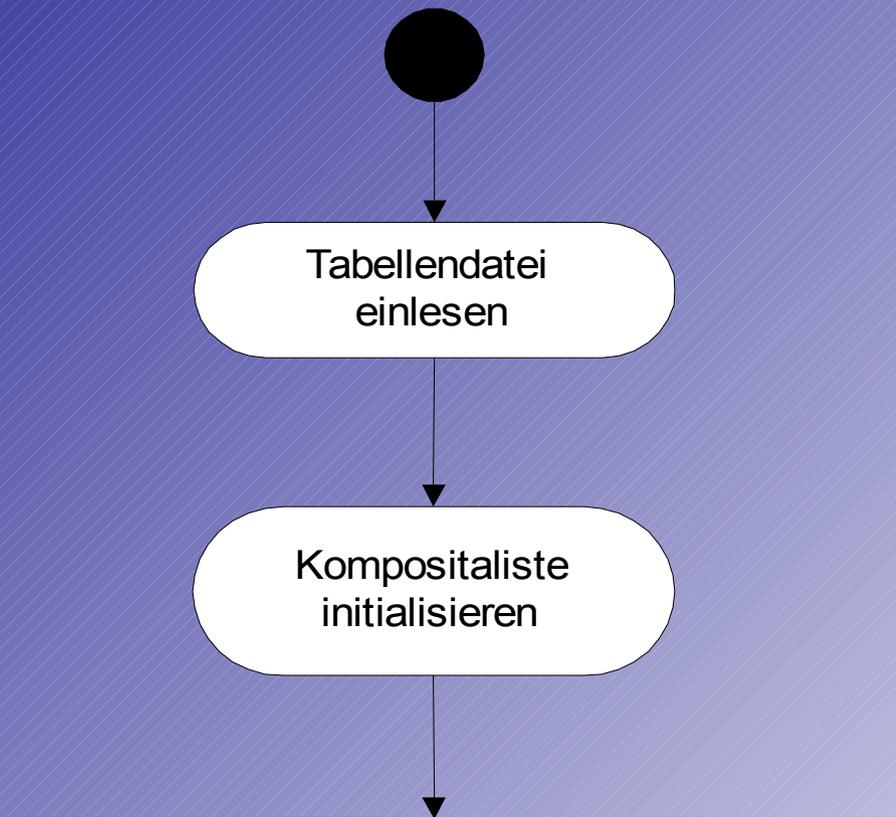


Datenbankzugriffsklasse

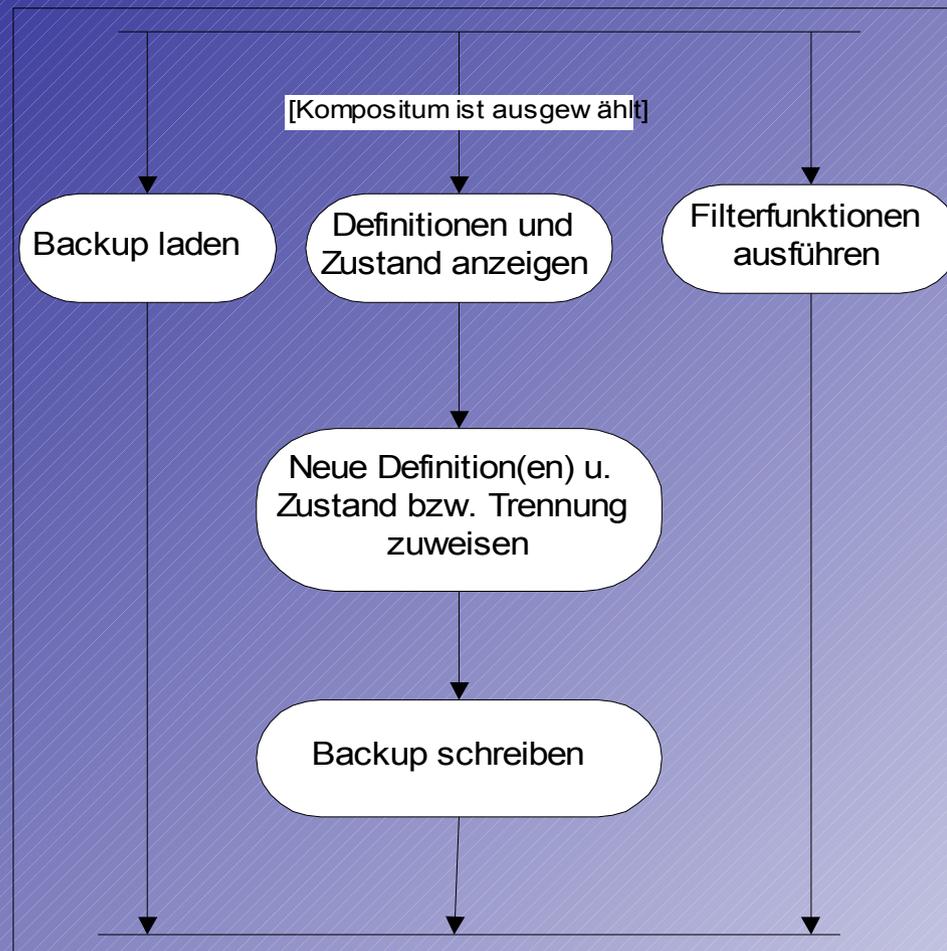


Programmablauf

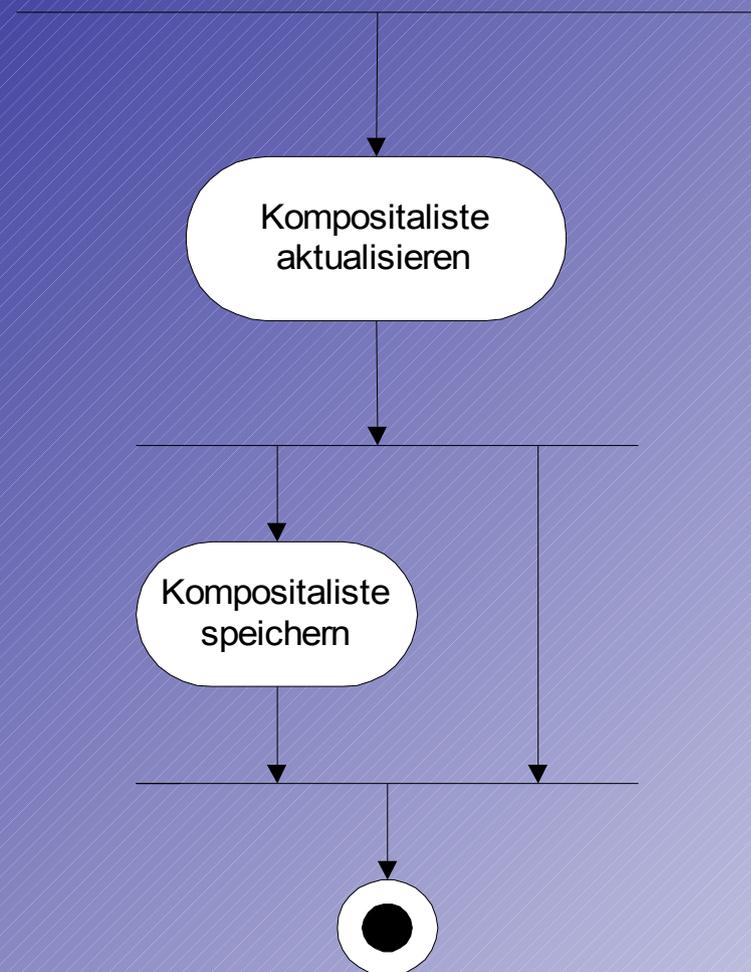
Aktivitätsdiagramm 1



Aktivitätsdiagramm 2



Aktivitätsdiagramm 3



Ziel erreicht?

- In der Endphase der Entwicklung genaue Abstimmungen mit dem Auftraggeber
- Die am Anfang und im Laufe des Projektes besprochenen Funktionalitäten sind realisiert worden
- TeSeT ist seit Juli 2003 im Einsatz
- Positive Rückmeldungen von Benutzern
- Einzelne Fehlerfälle tauchen auf, auch nach der Abgabe

Fehler

- Anforderungen wurden am Anfang nicht genau genug klargestellt und analysiert
- Es wurde zu "eifrig" programmiert. Eine Überarbeitung des Codes in kürzeren Abständen wäre nötig gewesen:
 - Design optimieren
 - Kommentieren

Folgen

- Das Projekt hat zu lange gedauert, vor allem am Anfang, als nach und nach Anforderungen klargestellt wurden
- Zum Schluss ist viel Arbeit nötig, um den Code zu überarbeiten und kommentieren, damit er von anderen weitergepflegt werden kann.

WWW.

- annanik.de
- ipsi.fraunhofer.de
- bifab.de
- cincom.com/products/smalltalk
- smalltalk.org
- goodstart.com [Smalltalk]