

Erweiterung einer Enterprise 2.0 Plattform um Funktionen für die Sichtbarkeit von Benutzeraktivitäten

Abschlussvortrag Bachelorarbeit

Rainer Niedermayr

Software Engineering betrieblicher Informationssysteme (sebis)

Ernst Denert-Stiftungslehrstuhl

www.matthes.in.tum.de

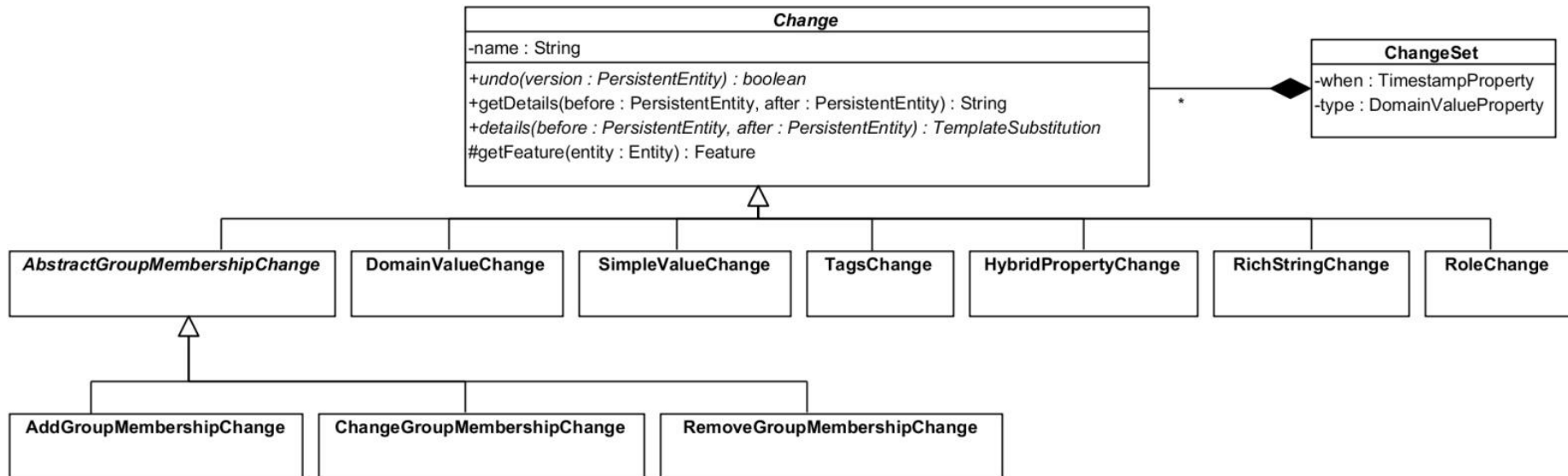
- Erweiterung von Tricia um Awareness-Funktionalitäten
 - Benutzer können auf dem Laufenden bleiben
 - Benutzer werden bei Änderungen informiert
 - Benutzer wissen, woran die anderen arbeiten

- Vorbereitende Maßnahmen
 - Verbesserung der Darstellung der elementaren Änderungen
 - Zusammenfassen von ähnlichen Änderungen

- Eigentliche Awareness-Funktionalitäten
 - Objekte beobachten
 - Auflistung der letzten Änderungen: Dashboard, Recent-Changes-Seite
 - E-Mail-Benachrichtigungen

Darstellung von elementaren Änderungen (1)

- Jede Bearbeitung eines Benutzers erzeugt ein ChangeSet



Change of feature "Tags"

Tags	
course	>> course
eist	eist
teaching	teaching
2011	tum
2010	sebis
software engineering	2011
uml	software engineering
	uml

Type Tags

lecture

Change: TAGS

+ Added tags

- sebis
- tum

- Removed tag

- 2010

+ Added type tag

- lecture

Change: TAGS

+ Added tags

- sebis
- tum

- Removed tag

- 2010

+ Added type tag

- lecture

- Demo

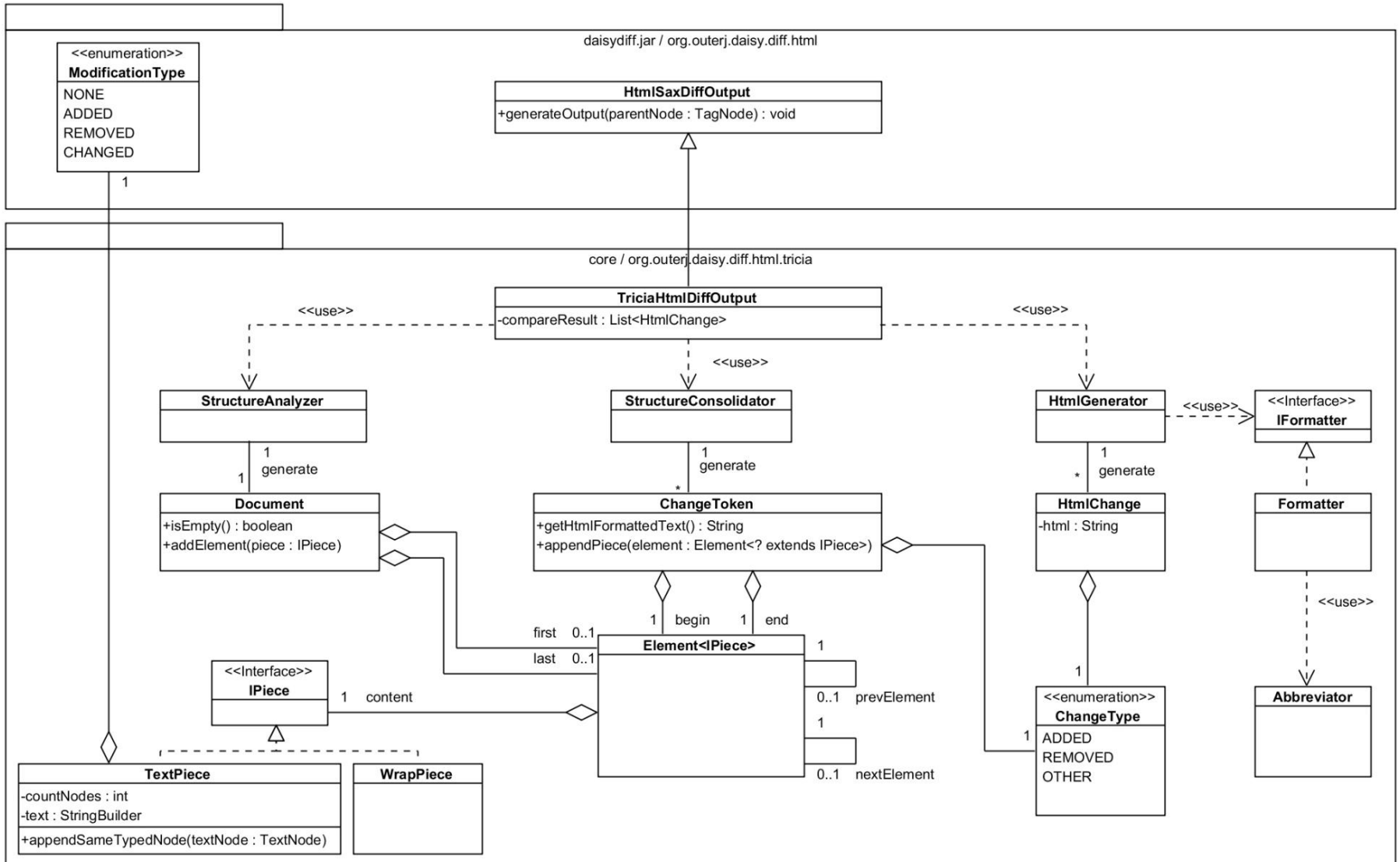
TriciaDiff (1)

- Vergleich von HTML-Dateien
- Verwendet für Änderungen von Rich-String-Eigenschaften
- Baut auf DaisyDiff auf

- **Eigenschaften**
 - Fokus auf die Änderungen
 - Darstellung geänderter Textstücke mit Kontext
 - Vorschau von langen Abschnitten
 - Entfernung der Originalformatierung

- 3-stufiger Vorgang
 - Eingabe umstrukturieren, Änderungen analysieren
 - Naheliegende Blöcke zusammenfassen
 - Blöcke kürzen, Kontext berechnen, HTML generieren

TriciaDiff (3)



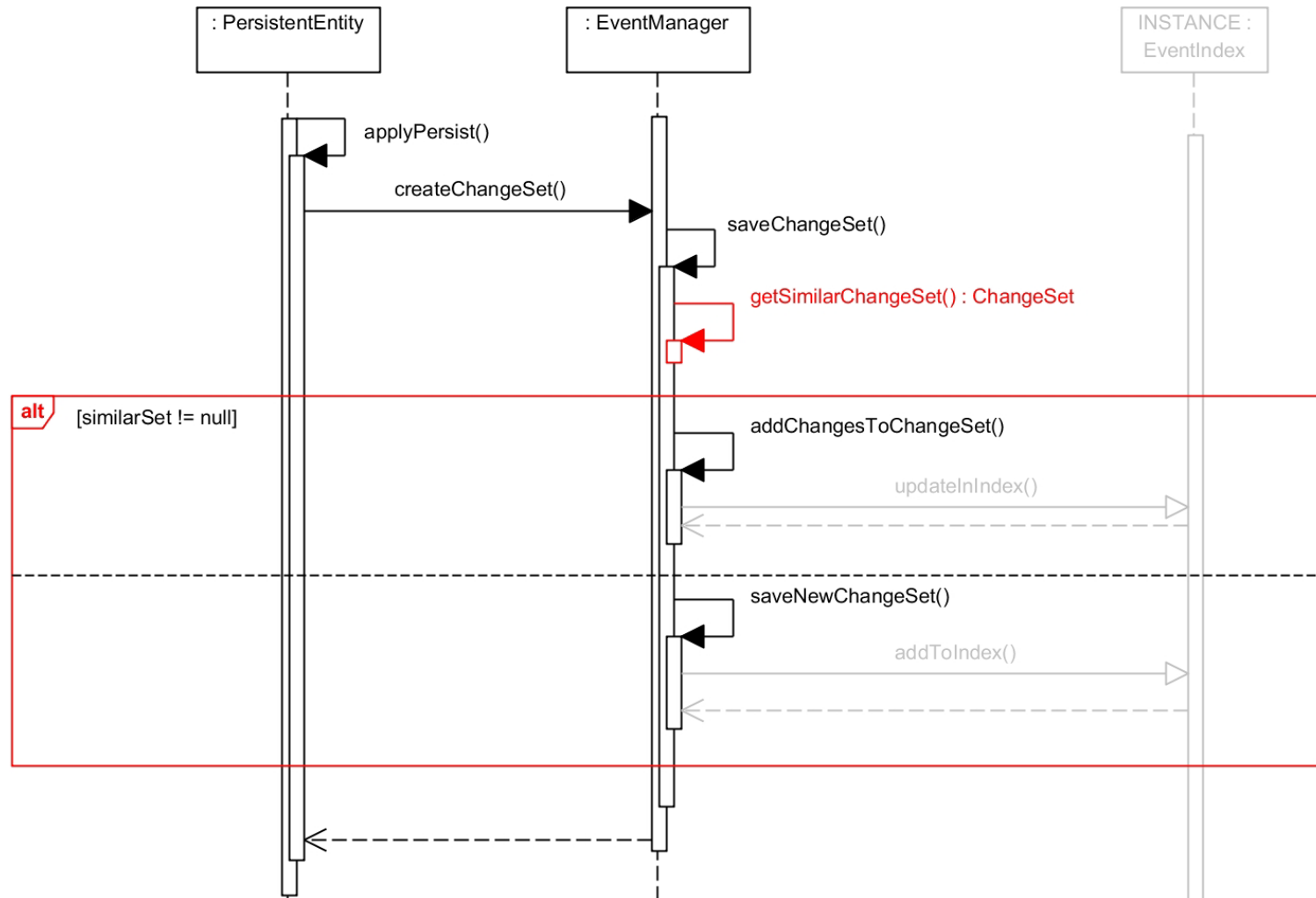
- Demo

- In-place-Bearbeitungen führen zu vielen Änderungen
- Versionshistorie wird unübersichtlich

→ ähnliche Änderungen aggregieren

- Änderungen gelten als ähnlich, wenn sie
 - die **gleiche Entität** betreffen und
 - vom **gleichen Benutzer** durchgeführt wurden und
 - weniger als **10 Minuten** auseinanderliegen und
 - den Typ **_edit** haben

Zusammenfassen von ähnlichen Änderungen (2)



- Codeausschnitt: ChangeSet mergen

```
private static void addChangesToChangeSet(ChangeSet changeSet, List<Change> newChanges, PersistentEntity entity) {

    PersistentEntity entityCopy = entity.createWritableCopy();
    Set<String> changedFeatures = Sets.newHashSet();

    undoChanges(entityCopy, newChanges, changedFeatures);
    undoChanges(entityCopy, changeSet.getChanges(), changedFeatures);







    List<Change> mergedChanges = ChangeSet.getDifferences(entity, entityCopy, changedFeatures);

    if (!mergedChanges.isEmpty()) {
        ChangeSet updatedChangeSet = changeSet.createWritableCopy();
        updatedChangeSet.adapt(ReadProtected.class).inherit(entity);
        updatedChangeSet.when.setToNow();
        updatedChangeSet.changes.set(ChangeSet.serialize(mergedChanges));
        updatedChangeSet.persist();

        EventIndex.INSTANCE().updateInIndex(updatedChangeSet);
    }
    else {
        changeSet.remove();

        EventIndex.INSTANCE().removeFromIndex(changeSet);
    }
}
```

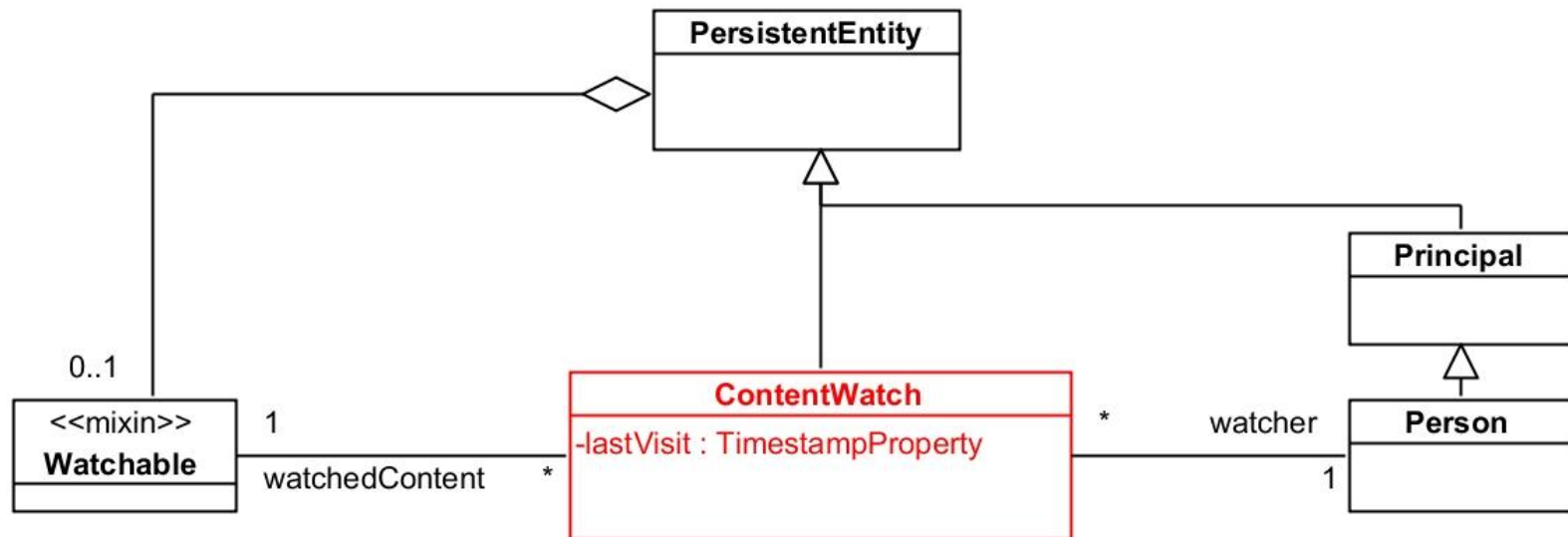
- Demo

- Beobachtbare Content-Typen
 -  Wikiseiten
 -  Wikis*
 -  Blogs*
 -  Benutzergruppen
 -  Attachments
 -  Directories*

* enthaltene Objekte dieser Spaces werden implizit auch beobachtet

Objekte beobachten (2)

- Datenmodell



- **Dashboard**

- Enthält die Liste der beobachteten Objekte
- Zeigt die letzten Änderungen dieser Objekte

- **Recent Changes**

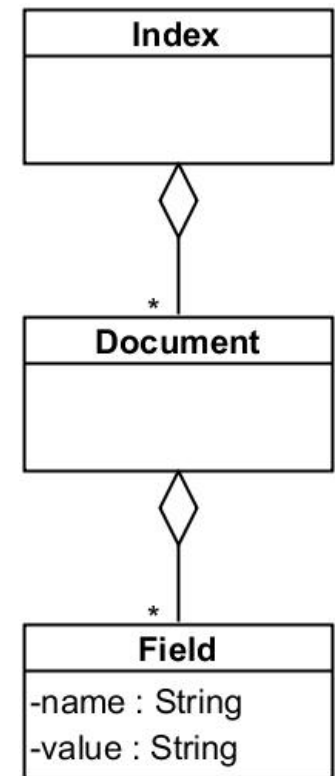
- Zeigt die letzte Änderung aller zugreifbaren Objekte an
- Bietet Filtermöglichkeiten

- Demo

Dashboard, Recent Changes (3)

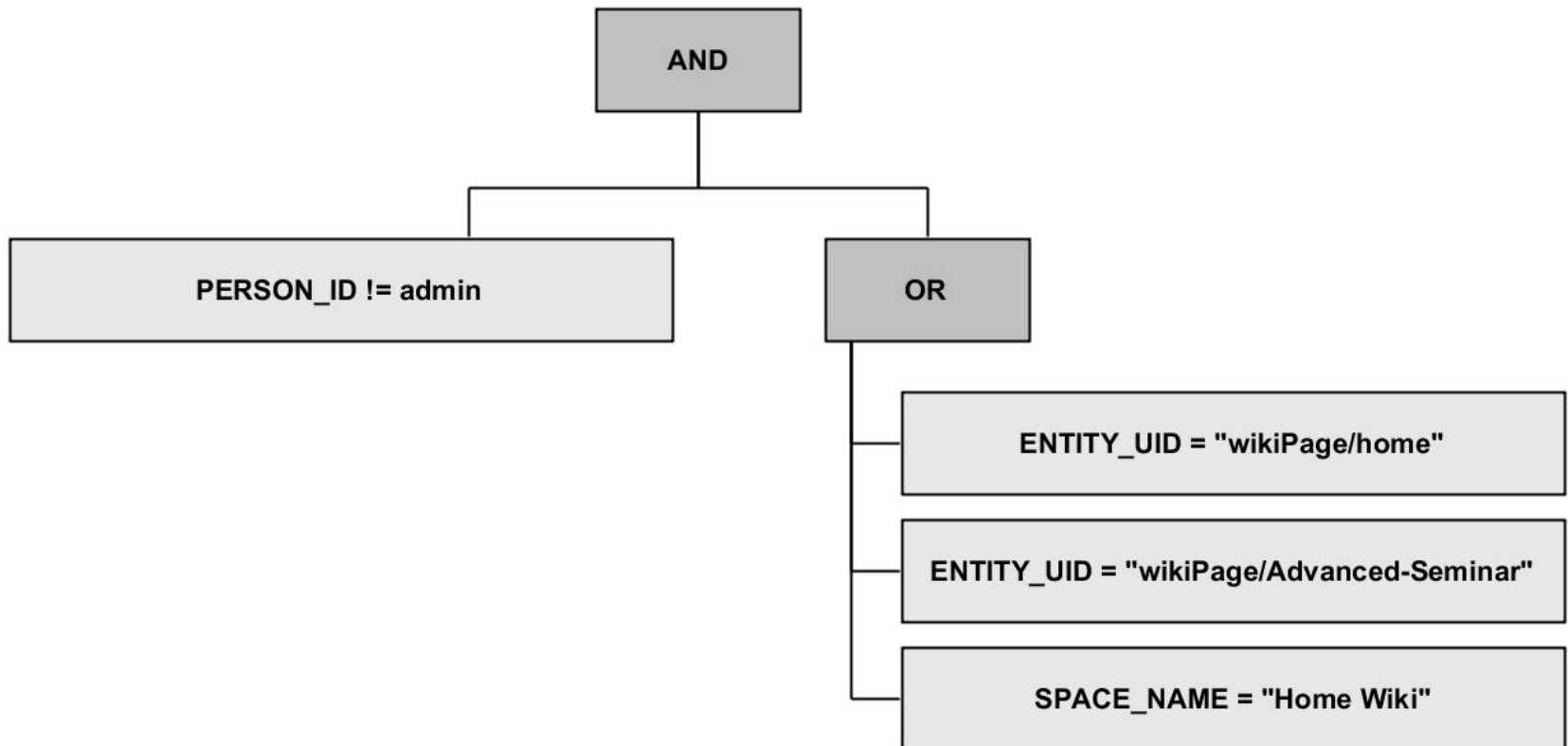
- Realisierung:
 - Indizierung von ChangeSets in einem Lucene-Index
 - Handler stellen Ergebnis der Indexabfragen aufbereitet dar

Event	
CHANGESET_ID	dc3opzocjpi6
EVENT_TYPE	_edit
ENTITY_UID	wikiPage/home
ENTITY_NAME	Wiki Page: Home
SPACE_NAME	Home Wiki
READ_ACCESS	everybody
PERSON_ID	admin
TIMESTAMP_FOR_SORTING	20110822084147
TAGS	main sebis



Dashboard, Recent Changes (4)

- Beispiel-Query für das Dashboard



Dashboard, Recent Changes (5)

- Codeausschnitt: Hinzufügen einer Query für beobachtete Objekte

```
public void addConstraintWatchedObjects(Person person) {
    BooleanQuery bQ = new BooleanQuery();

    for (ContentWatch currentWatch : person.watches.getAssets()) {
        Entity watchedObject = currentWatch.watchedObject.getEntity();

        Term t;

        if (watchedObject.hasMixin(Space.class)) {
            t = new Term(Event.FIELD_SPACE_NAME, watchedObject.adapt(Space.class).getName());
        } else {
            t = new Term(Event.FIELD_ENTITY_UID, watchedObject.getUid());
        }

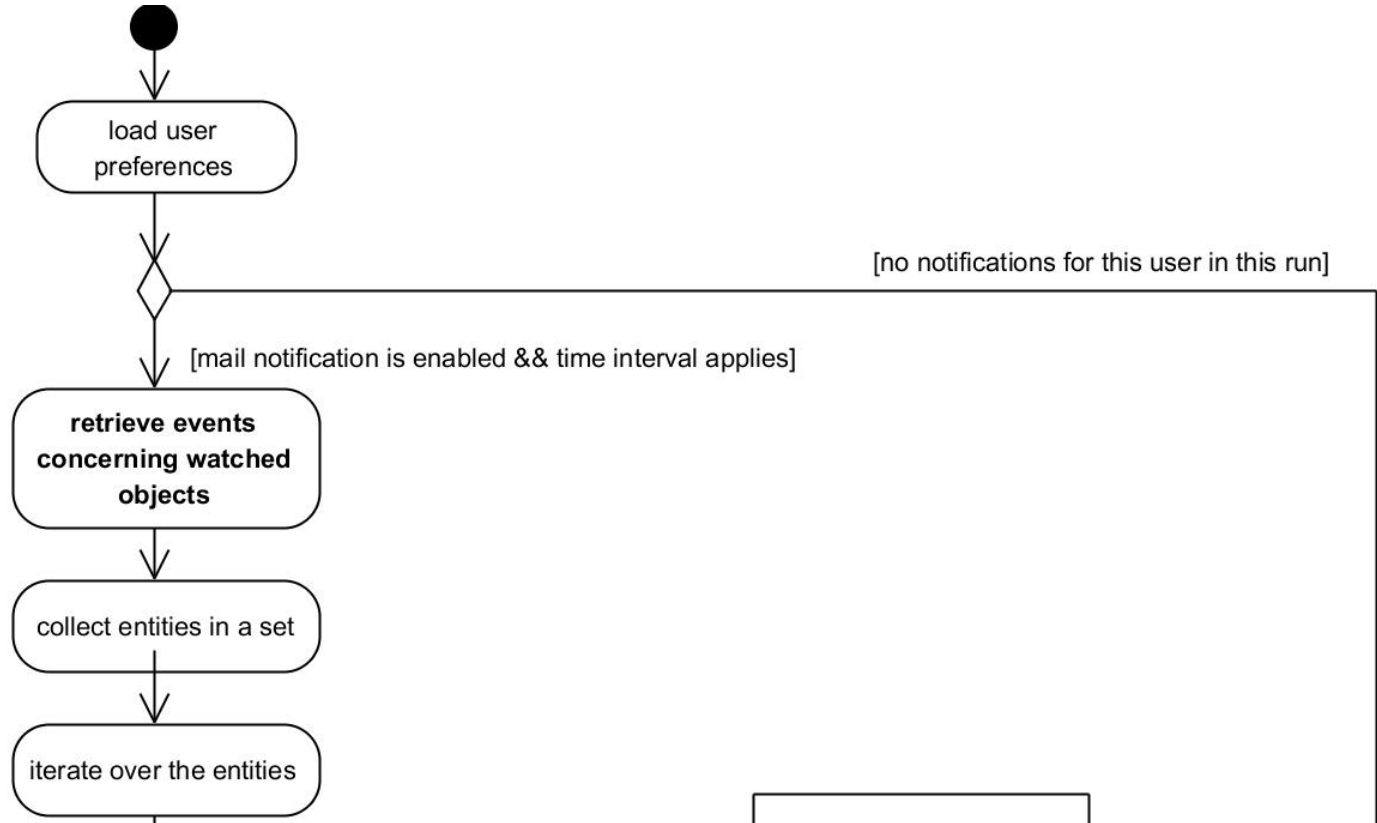
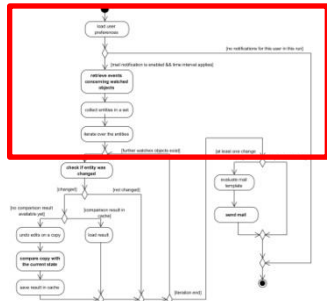
        TermQuery tQ = new TermQuery(t);
        bQ.add(tQ, Occur.SHOULD);
    }

    addSubQuery(bQ);
}
```

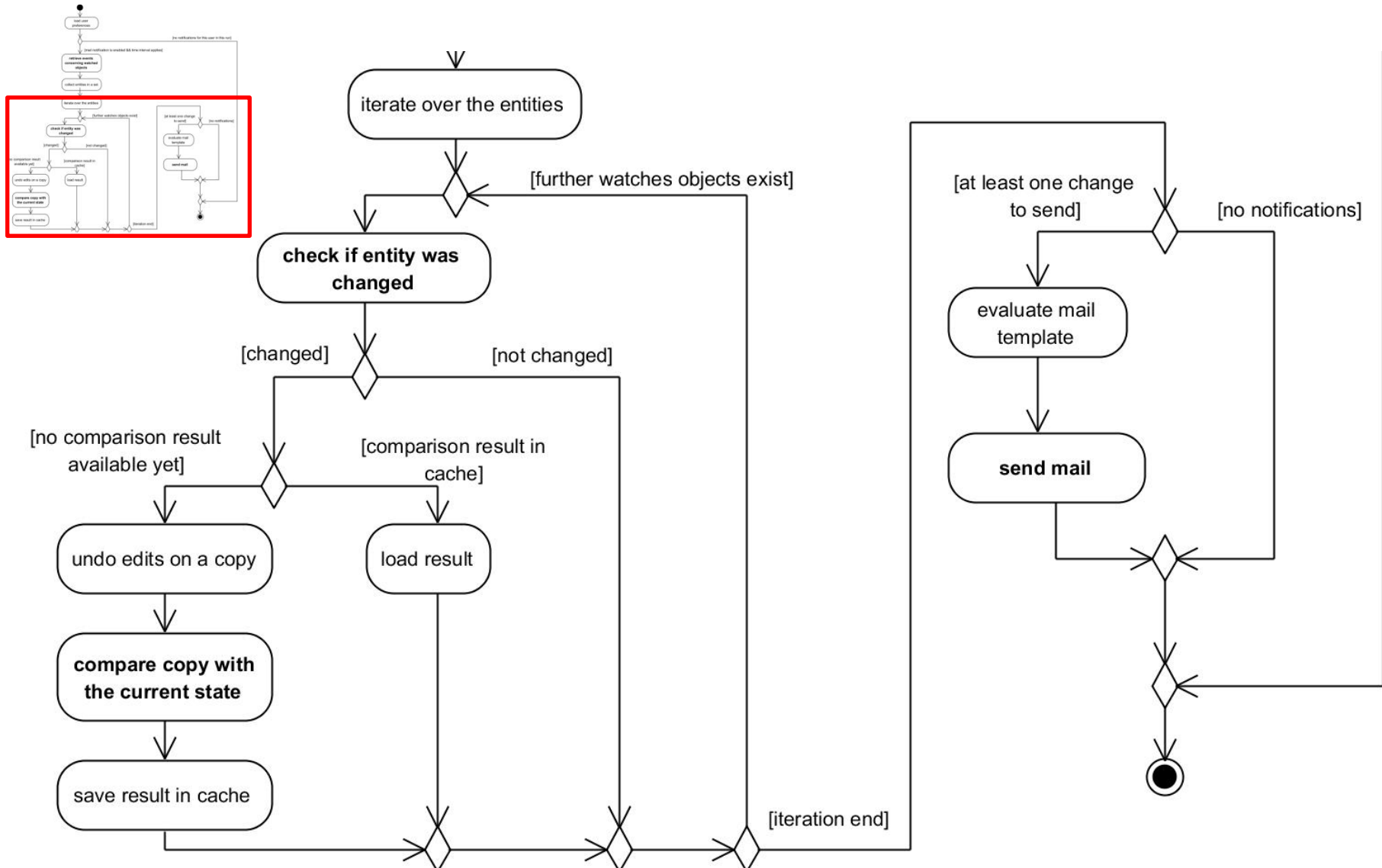
- **Idee:** regelmäßige versendete E-Mails fassen Änderungen beobachteter Objekte zusammen
 - z. B. alle beobachteten Änderungen eines Tages

- Vom Benutzer konfigurierbar
 - Benachrichtigungsdienst aktiv / inaktiv
 - Zeitintervall
 - Granularität des Inhalts

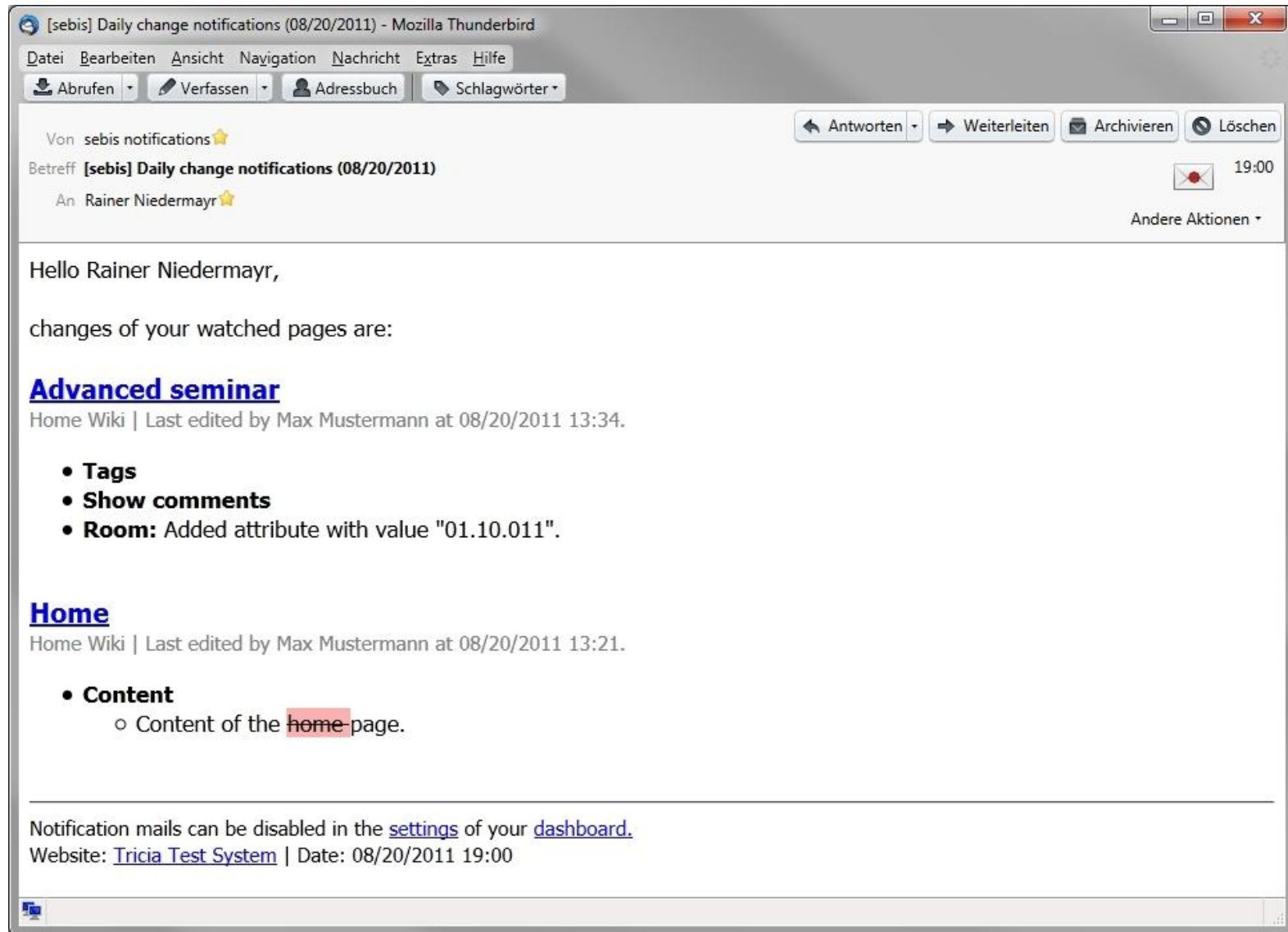
E-Mail-Benachrichtigungen (2)



E-Mail-Benachrichtigungen (3)



E-Mail-Benachrichtigungen (4)



Danke für die Aufmerksamkeit.