powered by Cha-Q

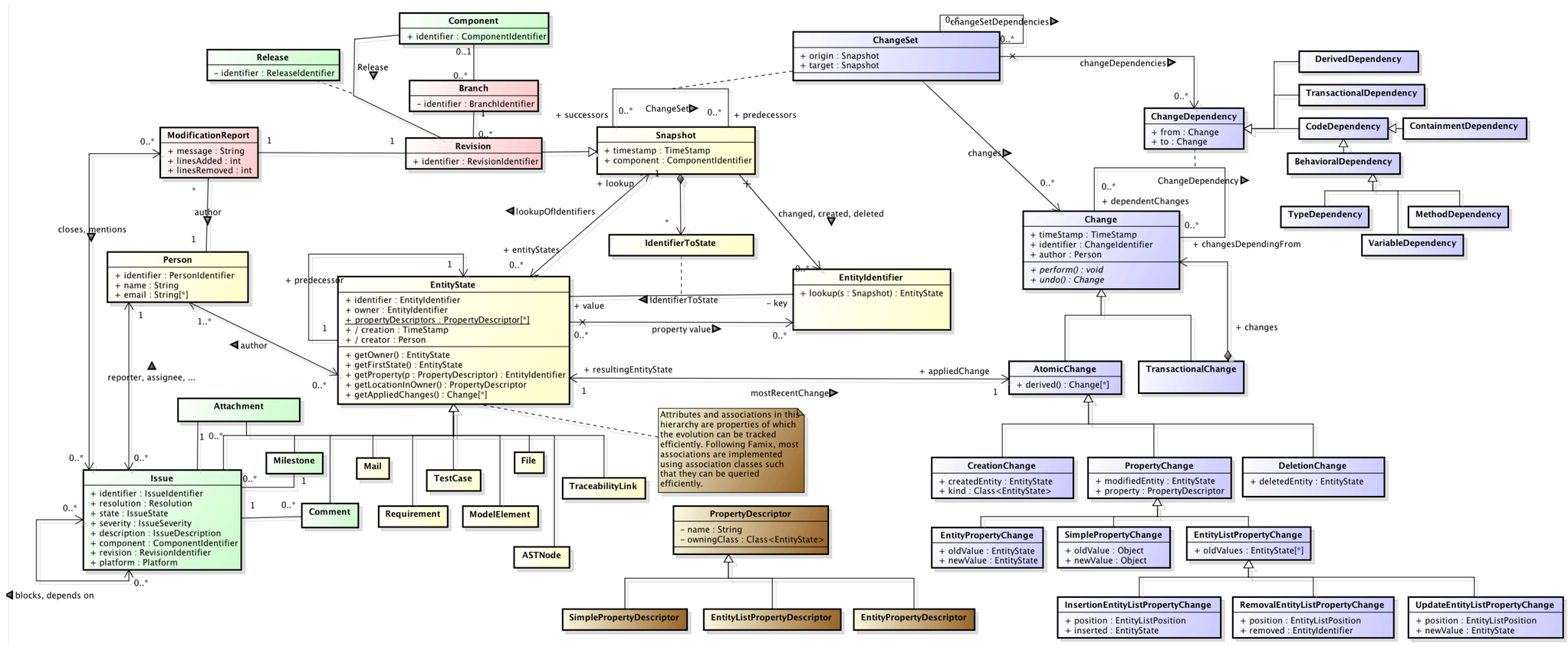
A Change-Centric Software Representation

Research on Analysing changes

The Cha-Q meta-model

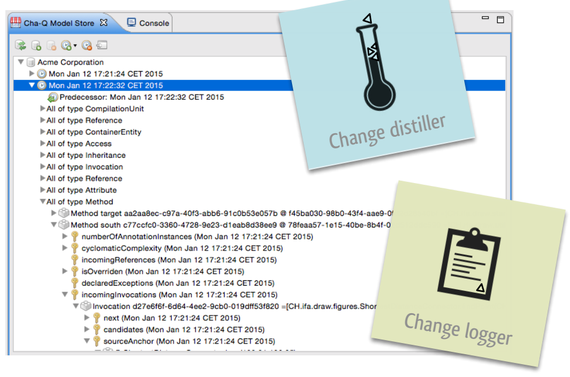
First interconnected representation of:

- ✓ State & evolution of software entities
- ✓ Changes to entities
- ✓ System snapshots under version control

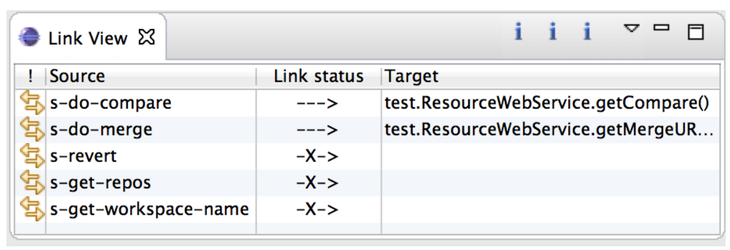


Applications of the meta-model

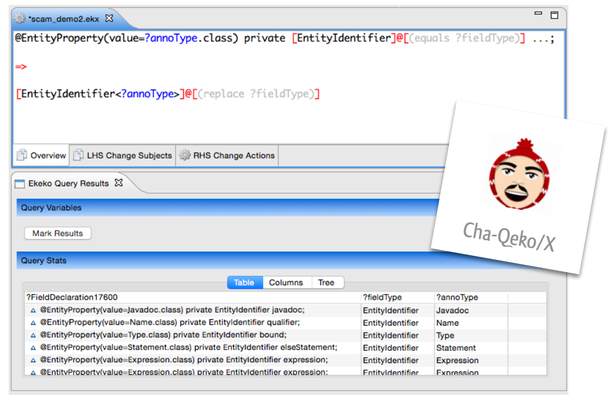
Analysing changes



Maintaining tracability links



Automating repeated changes



Memory-efficient state tracking

```
public int addNumber(int x, int y) {
    return x+y;
}
```

- Method id: M1
  - Name (id: N1)
  - Body (id: E1)
  - TypeParameters (id: L1)
  - ...

Snapshot 1: M1 N1 E1 L1 ...

```
public int addInt(int x, int y) {
    return x+y;
}
```

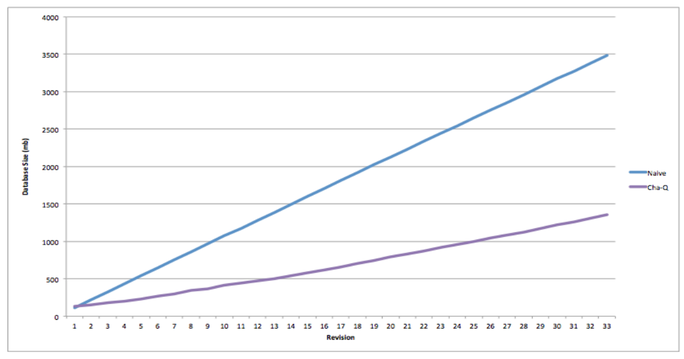
Only changes are stored

- Method id: M1
  - Name (id: N2)
  - Body (id: E1)
  - TypeParameters (id: L1)
  - ...

Snapshot 2: M1 N2 E1 L1 ...

Disk footprint evaluation

Neo4j + weak references



Evolution of Exapus project

Single revision: 194149 entities, 223979 properties, and 194147 relationships of 32 distinct types

On average: 22,5 files per revision changed

Highlights

- ✓ Change-centric representation of code, issues, releases, ...
- ✓ Memory-efficient
- ✓ Extensible & language-agnostic

Importers for:

- Java source code
- Bugzilla & JIRA issue trackers
- XML files

The Implementation of the Cha-Q Meta-Model:
A Comprehensive, Change-Centric Software Representation
Coen De Roover, Cristophe Scholliers, Viviane Jonckers, Javier Pérez, Alessandro Murgia, Serge Demeyer
Electronic Communication of the European Association of Software Science and Technology, Volume 65 (2014)

http://soft.vub.ac.be/chaq

Universiteit Antwerpen

Vrije Universiteit Brussel

iMT