

Link Reconstruction: A Forward Engineering Concern

Arie van Deursen

CWI & TU Delft

Observation I

- Companies need information in terms of requirements
 - Progress monitoring and reporting
 - Status allocation (implemented, tested, ...)
 - Impact analysis – new feature may affect other features
- Maintaining traceability links manually is too cumbersome
 - Not done, hence info not used

Observation II

- Links can exist between a wealth of entities, such as
 - Features (*End-user visible characteristic of a system*)
 - Concerns (*Any matter of interest in a software system*)
 - Concepts (*Human conceptual knowledge as opposed to program structures*)
 - Requirements (*Description what a system should do*)
 - Source code, design decision, products, UML model, build task, release, use case, test case,

Observation III

- *Reconstruction == querying*
 - Pointcut language as in the *Concern Manipulation Environment*
 - Program dependence graph as it can be accessed, e.g., in Lisp via CodeSurfer plugin
 - Information retrieval as in, e.g., latent semantic indexing

What are the structures we are querying?
What are the query languages we are using?

Observation IV

- *Thinking ahead will ease later querying*
- Annotations
- Naming and coding conventions
- Establish a proper structure for your requirements documents
 - Explicit meta-model (as in, e.g., Doors)

What should be done by hand?
How can we minimize the effort needed?
How can we ensure integrity?