

Visualizing Non-Functional Requirements Patterns

Sam Supakkul and Lawrence Chung

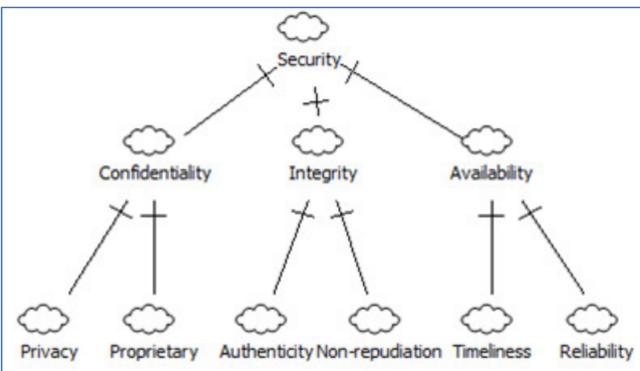
Motivation

- Knowledge of NFRs can have complex structure and relationships
- The structure and relationships are hard to see when knowledge is captured using text
- Captured knowledge should be conveniently reusable during requirements engineering

Objectives

- Knowledge of NFRs captured as visual patterns
- NFR patterns captured, organized, and reused in a tool-assisted environment

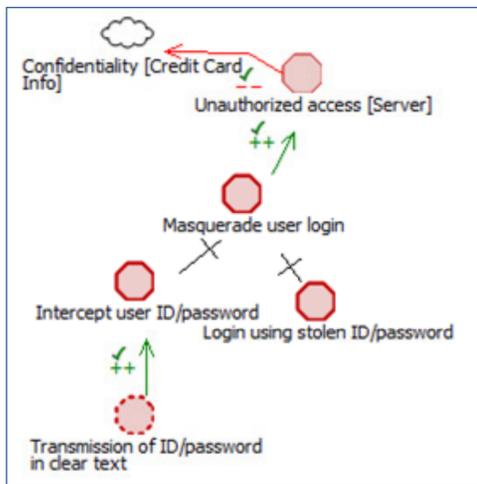
NFR Patterns



An objective pattern captures a definition of an NFR as a softgoal and its sub-goals to be achieved

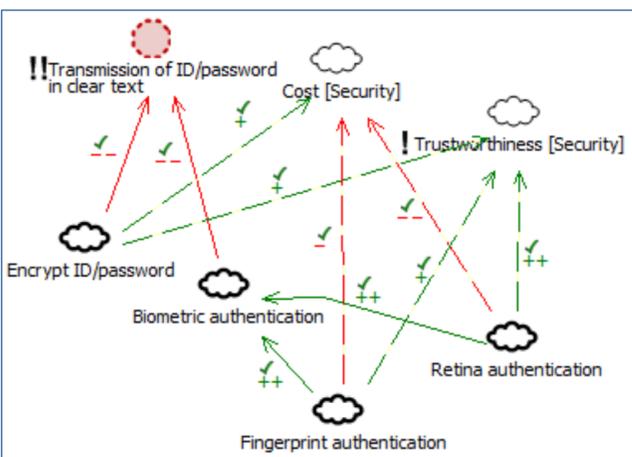
Objective Pattern

A problem pattern captures obstacles and their causes that hurt the achievement of a goal.



Problem Pattern

An alternatives pattern captures alternative means to achieving a goal or solutions to mitigating a problem.

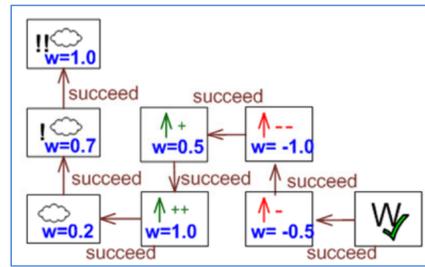


Alternatives Pattern

Quantitative Weight-based Selection
 $= \max(\sum \text{weight}(\text{criticality}) \times \text{weight}(\text{contribution}))$

Qualitative Rank-based Selection
 $= \text{best}(\sum \text{rank}(\text{contribution}, \text{criticality}))$

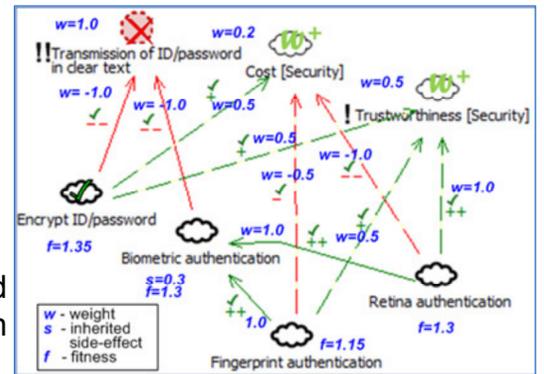
Quantitative and Qualitative Selection Rules



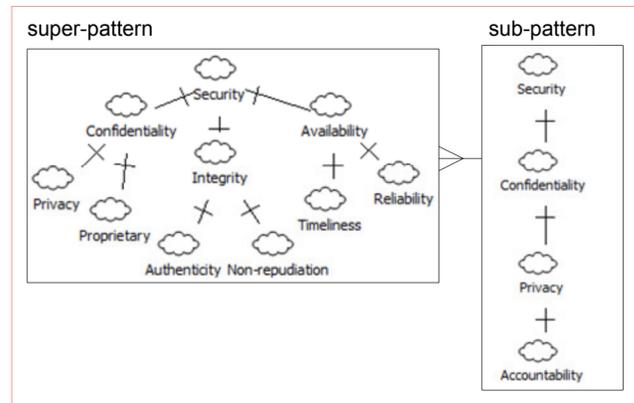
Selection Pattern

A selection pattern captures an application independent selection scheme, consisting of a series of weight/ranking assignments and a selection rule.

An example of a weight-based selection pattern application



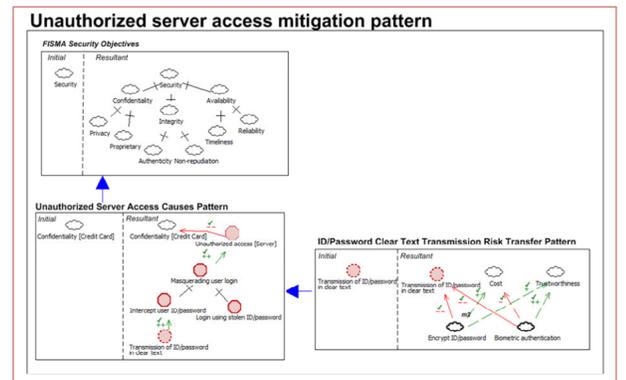
Pattern Organization



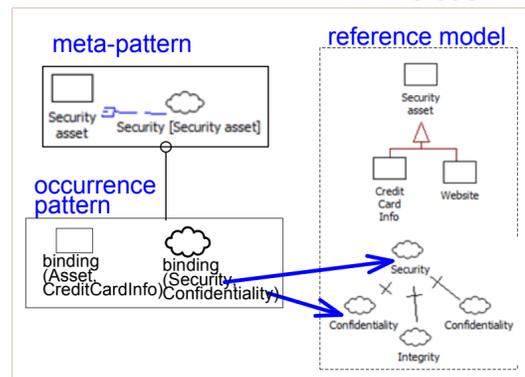
A specialized pattern captures a more specific definition of an NFR

Pattern Specialization

A composite pattern assembles smaller patterns to capture a larger chunk of knowledge



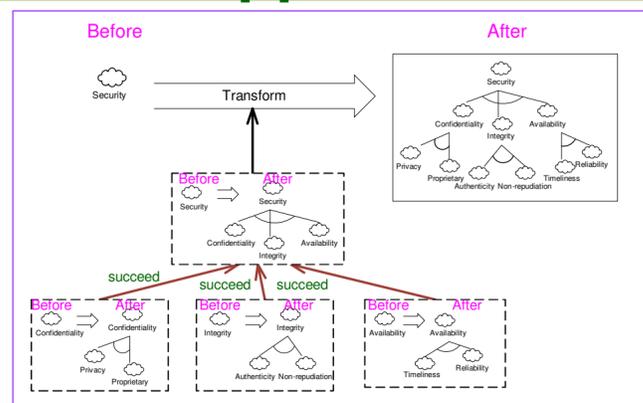
Pattern Composition



A pattern can be used as a template to create new patterns with customizations specified by the binding specifications

Pattern Instantiation

Tool Support



Each pattern is captured with single-step model refinement rules that can be applied by a tool to transform a target model when the pattern is applied

Model Transformation with Refinement Rules