

Modeling Non-Functional Requirements Using Soft Goals

Requirements Engineering II –
HS10

Contents

- Non-functional requirements
- NFR Framework
- Softgoal Interdependency Graph (SIG)
 - Idea
 - Elements
 - Notation
- Design patterns and softgoals

Non-functional requirements

- Specify criteria used to judge the operation of a system, not behavior
- Also called “qualities” of a system
 - Constraints
 - Quality attributes
 - Quality goals
 - ...
- Implementation plan from the system architecture

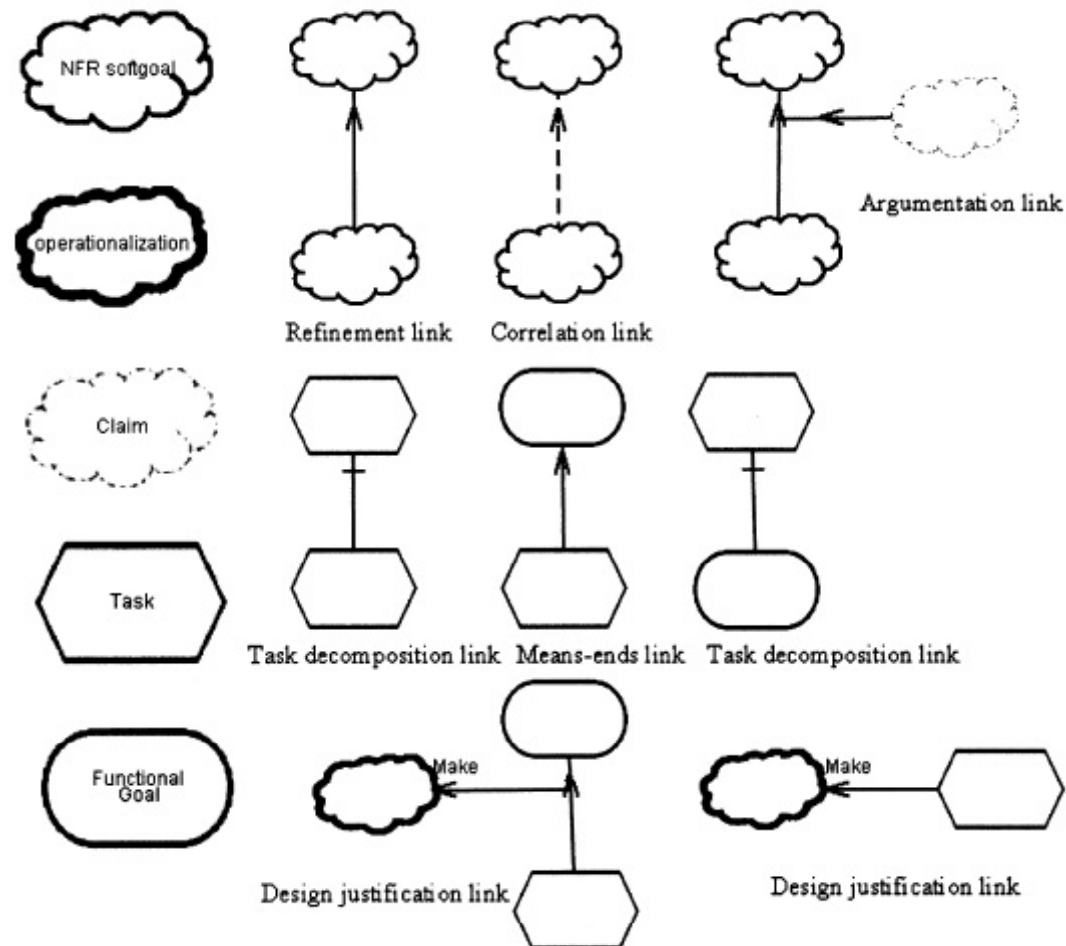
NFR Framework

- Catalogue of design knowledge
 - NFR types
 - Development techniques
 - Implicit interdependencies
- Steps in design process
 - Acquire
 - Identify
 - Decompose
 - Operationalization
- Software Interdependency Graphs

SIG - Idea

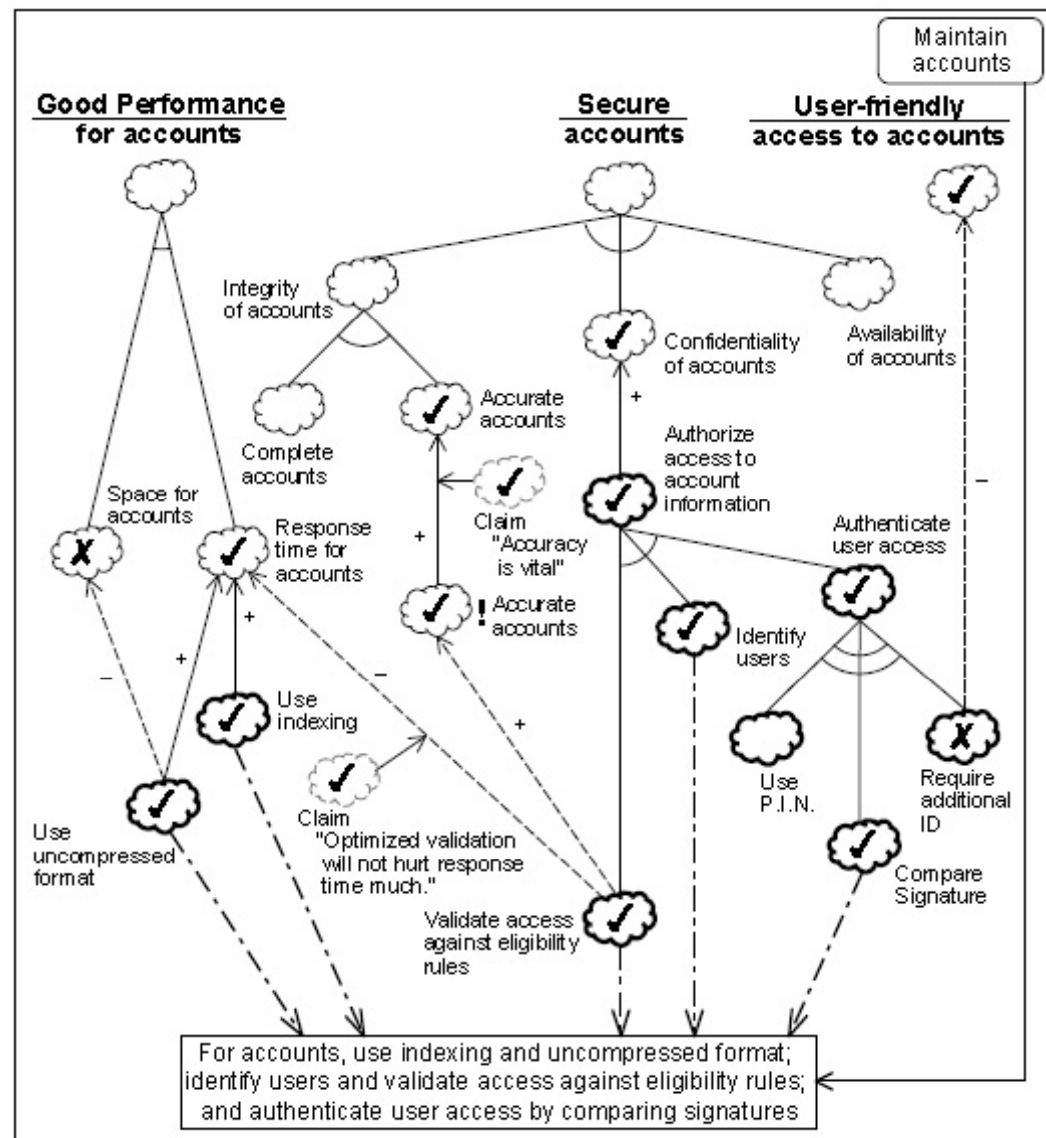
- Starting with high-level softgoals or nonfunctional requirements
- Break down softgoals into smaller softgoals (hierarchical)
- Detect ambiguities, interdependencies, domain information, priorities
- Choose alternatives , reason why
- Analyze results

SIG - Elements



Source: Gross , 2001

Example



Source: Chung, 2000

SIG - Problems

- Large systems
- Handling of SIG

Design patterns and soft goals

- Design patterns are generalized solutions to recurring problems
 - Solution to NFR included
 - Problems addressed, implementation tradeoffs
- Software design often uses design patterns to avoid re-inventing the wheel
- Analyzing addressed NFR in design patterns during design for meeting design requirements
=> meeting NFRs

Handling design patterns

- Analyze 3 sections of a pattern
 - Intent
 - Motivation
 - Structure
- Use the 4 steps described on slide 4
 - Acquire
 - Identify
 - Decompose
 - Operationalization

Benefits

- Explicit NFRs in design patterns help understand rationale of design
- Patterns easier to analyze
- Open possibility for tool support
- Better understand trade-offs during design
- Design history enables reasoning and understanding of development life cycle and system evolution