

Chapter 16

Rationale Management

Object-Oriented
Software Construction

Armin B. Cremers, Tobias Rho, Daniel Speicher & Holger Mügge
(based on Bruegge & Dutoit)



- What is rationale?
- Why is it critical in software engineering?
- Centralized traffic control example
- Rationale in project management
 - ◆ Consensus building
 - ◆ Consistency with goals (NFR Framework)
- Summary

What is rationale?

Rationale is the reasoning that led to the system.

Rationale includes:

- the **issues** that were addressed,
- the **alternatives** that were considered,
- the **decisions** that were made to resolve the issues,
- the **criteria** that were used to guide decisions, and
- the **debate** developers went through to reach a decision.

Why is rationale important in software engineering?

Many software systems are complex:

They result from a large number of decisions taken over an extended period of time.

- Evolving assumptions
- Legacy decisions
- Conflicting criteria

-> high maintenance cost

-> loss & rediscovery of information

- Improve design support
 - ◆ Avoid duplicate evaluation of poor alternatives
 - ◆ Make consistent and explicit trade-offs

- Improve documentation support
 - ◆ Makes it easier for non developers (e.g., managers, lawyers, technical writers) to review the design

- Improve maintenance support
 - ◆ Provide maintainers with design context

- Improve learning
 - ◆ New staff can learn the design by replaying the decisions that produced it

Argumentation is the most promising approach so far:

- More information than (design) documents: captures trade-offs and discarded alternatives that design documents do not.
- Less messy than communication records: communication records contain everything.

Issue models represent arguments:

- Nodes represent argument steps
- Links represent their relationships

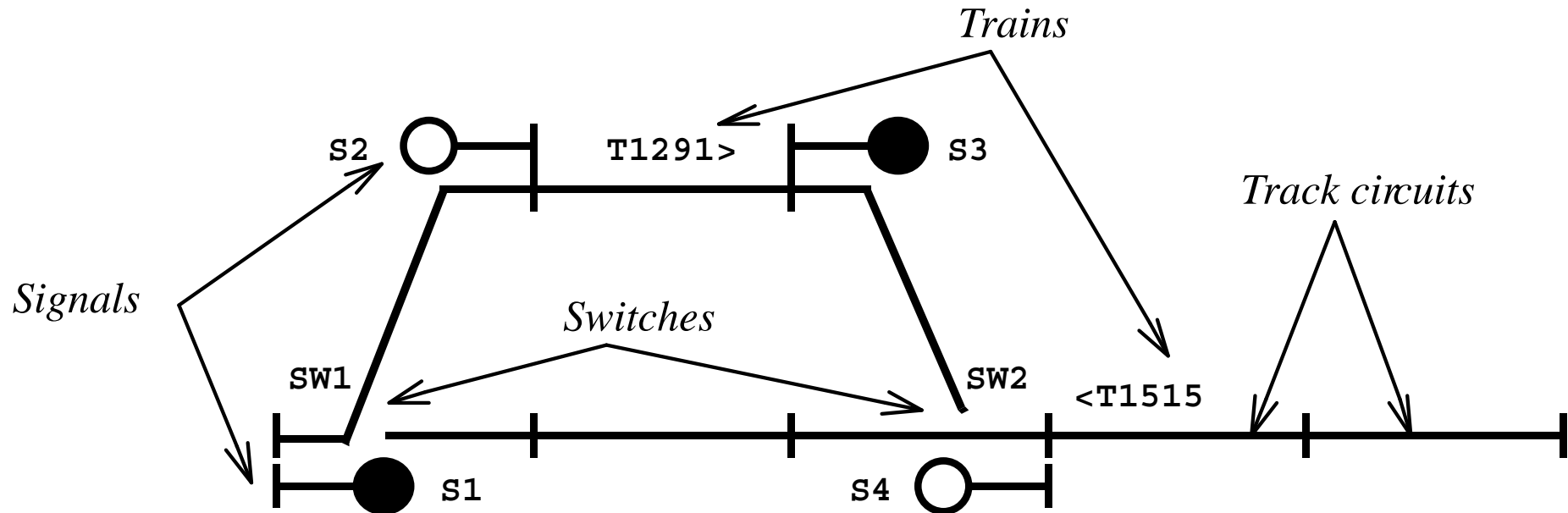
Question: Alternative Authentication Mechanisms?

References: Service: Authenticate

Decision: Smart Card + PIN

	Criteria 1: ATM Unit Cost	Criteria 2: Privacy
Option 1: Account number	+	-
Option 2: Finger print reader	-	+
Option 3: Smart Card + PIN	+	+

Centralized traffic control



- CTC systems enable dispatchers to monitor and control trains remotely
- CTC allows the planning of routes and replanning in case of problems

Centralized traffic control (2)

CTC systems are good examples of rationale capture:

- Long lived systems (some systems include relays installed last century)
 - ◆ Extended maintenance life cycle
- Although not life critical, downtime is expensive
 - ◆ Low tolerance for bugs
 - ◆ Transition to mature technology

- Issues are concrete problems which usually do not have a unique, correct solution.
- Issues are phrased as questions.

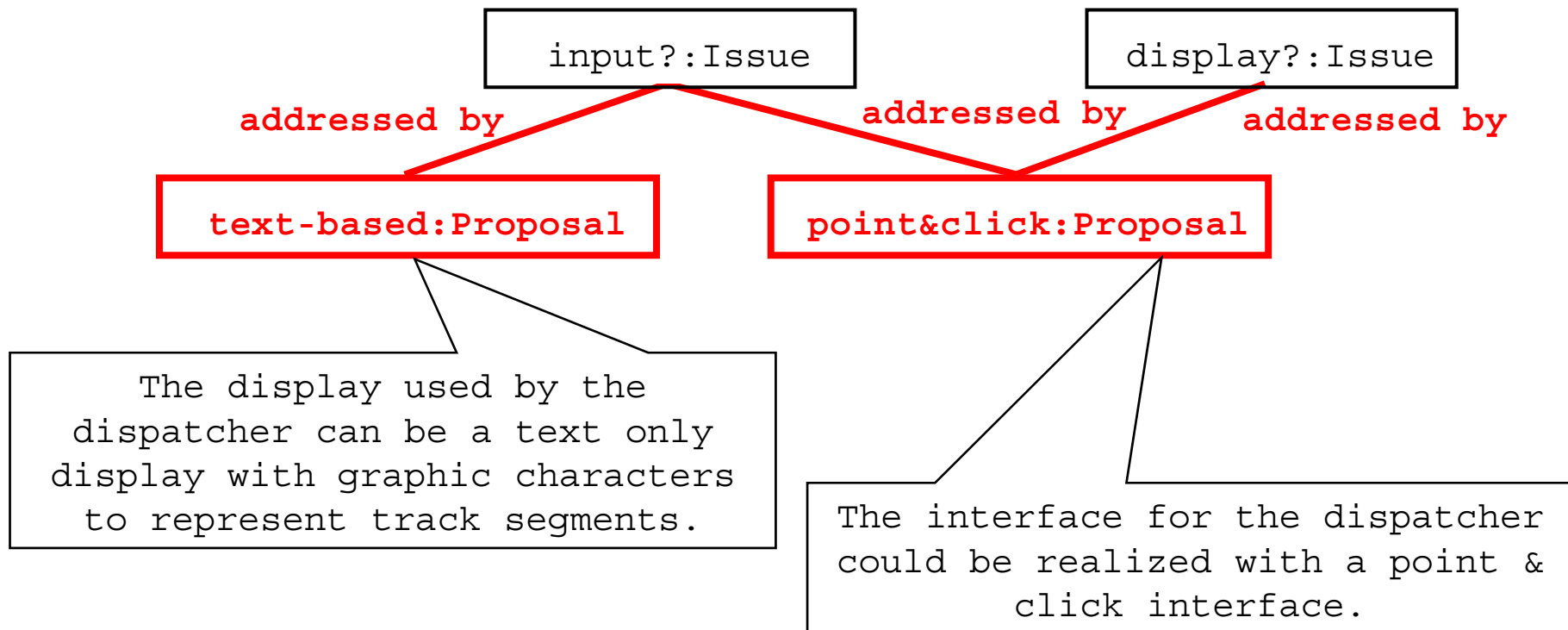
input?:Issue

How should the dispatcher
input commands?

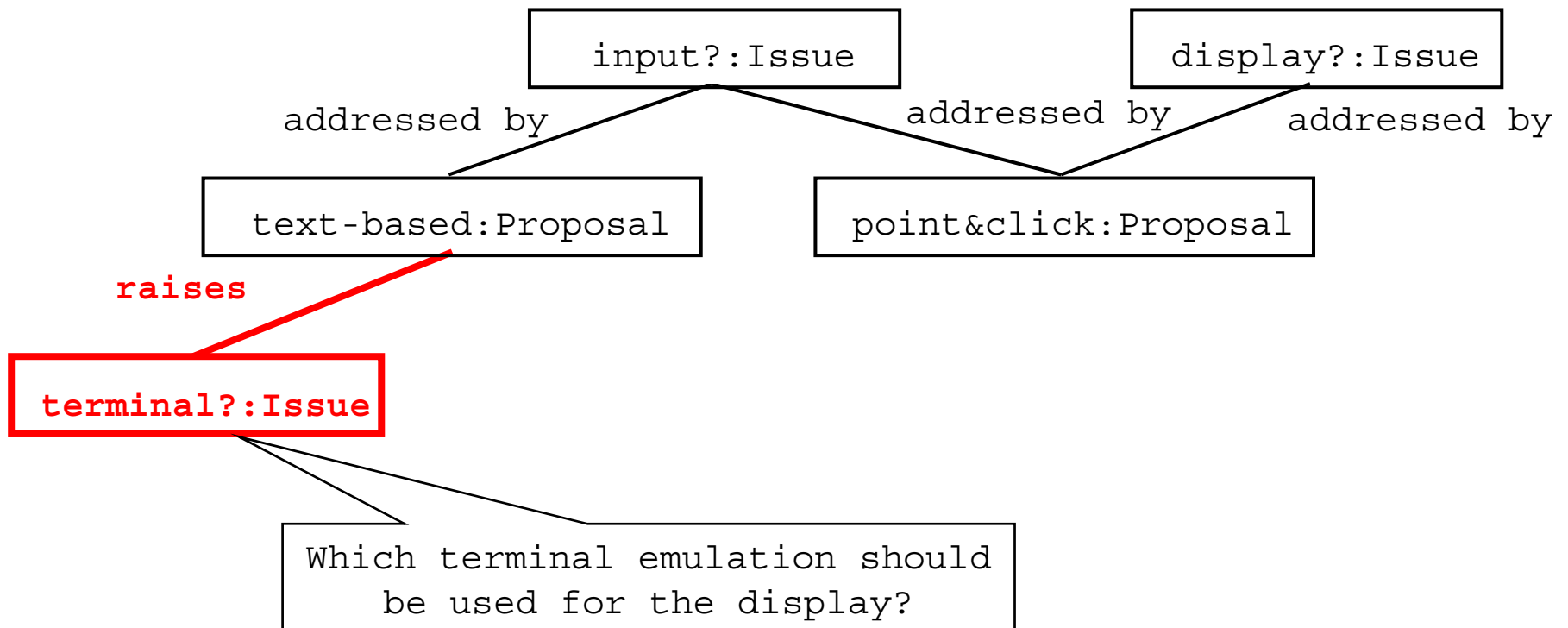
display?:Issue

How should track sections
be displayed?

- Proposals are possible alternatives to issues.
- One proposal can be shared across multiple issues.

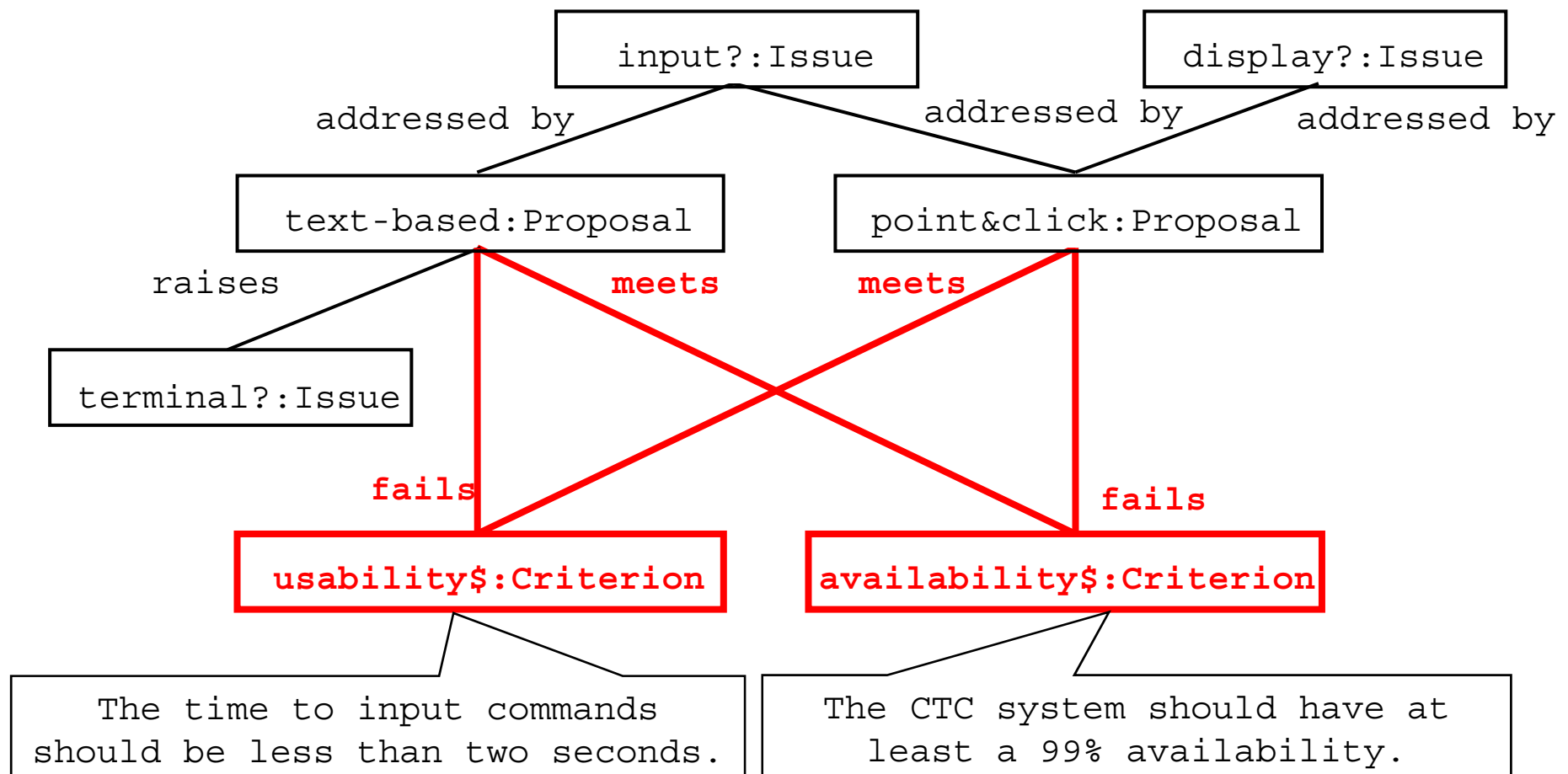


- Consequent issues are issues raised by the introduction of a proposal.



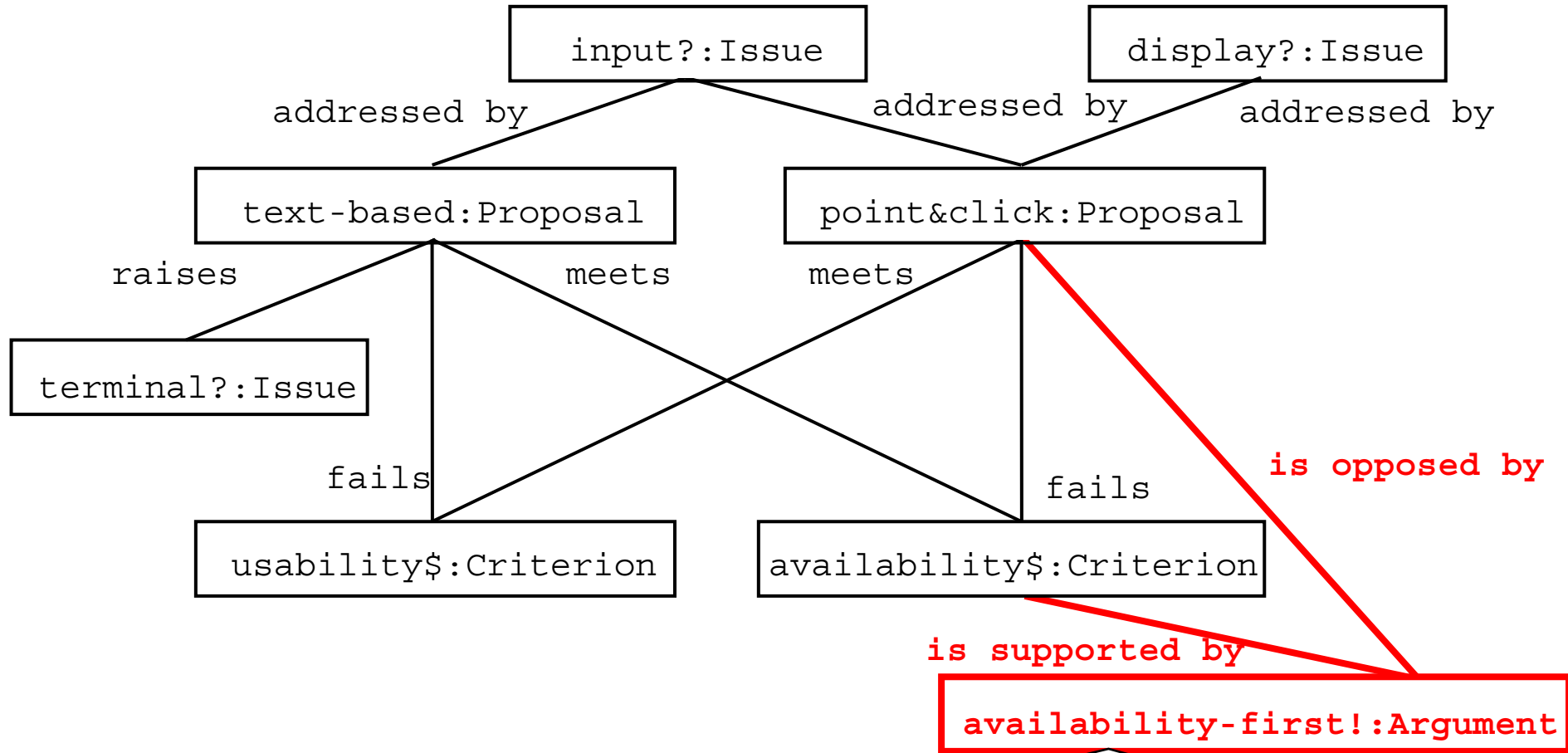
Criteria

- A criteria represent a goodness measure.
- Criteria are often design goals or nonfunctional requirements.



- Arguments represent the debate developers went through to arrive to resolve the issue.
- Arguments can support or oppose any other part of the rationale.
- Arguments constitute the most part of rationale.

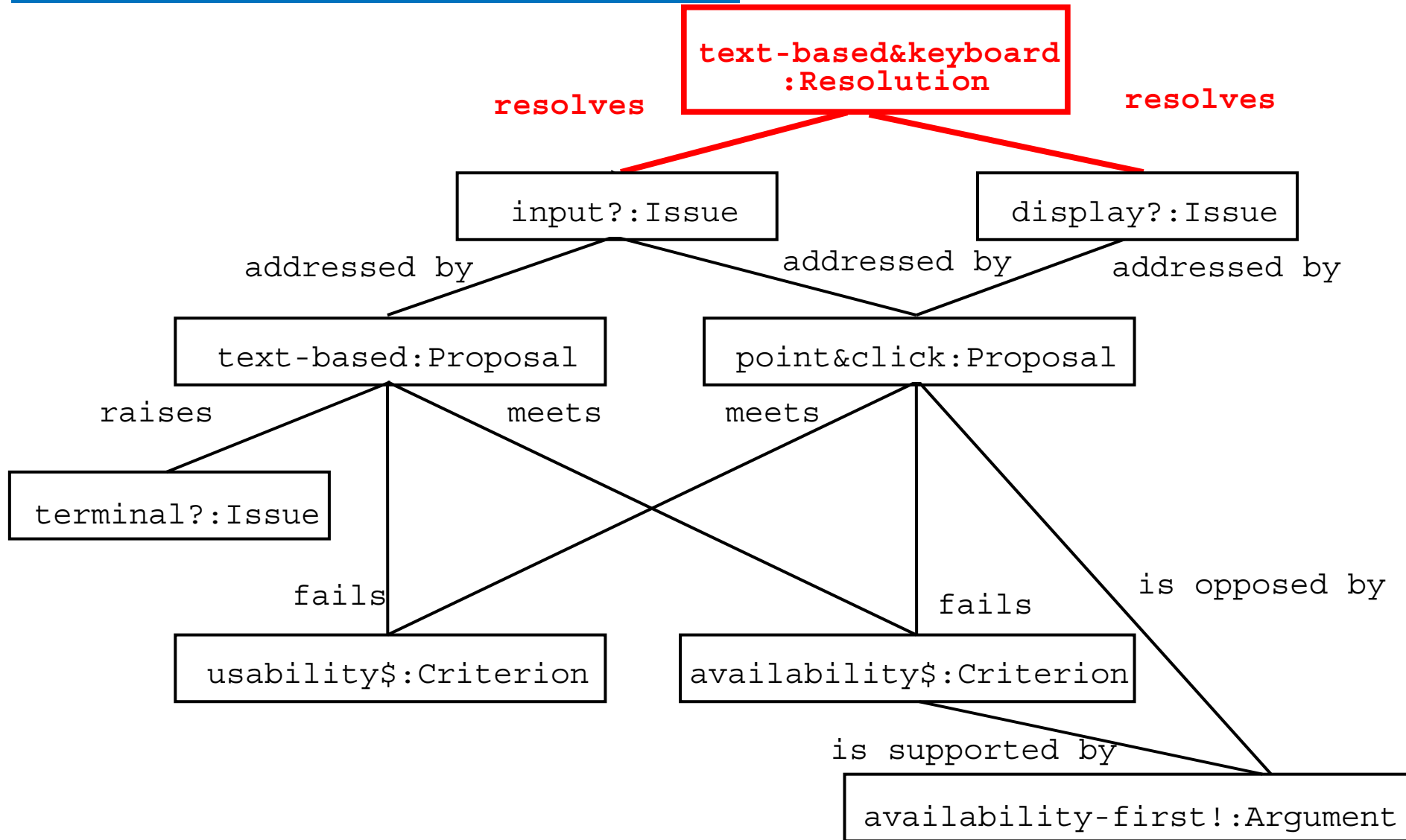
Arguments (2)



Point&click interfaces are more complex to implement than text-based interfaces. Hence, they are also more difficult to test. The point&click interface risks introducing fatal errors in the system that would offset any usability benefit the interface would provide.

- Resolutions represent decisions.
- A resolution summarizes the chosen alternative and the argument supporting it.
- A resolved issue is said to be closed.
- A resolved issue can be re-opened if necessary, in which case the resolution is demoted.

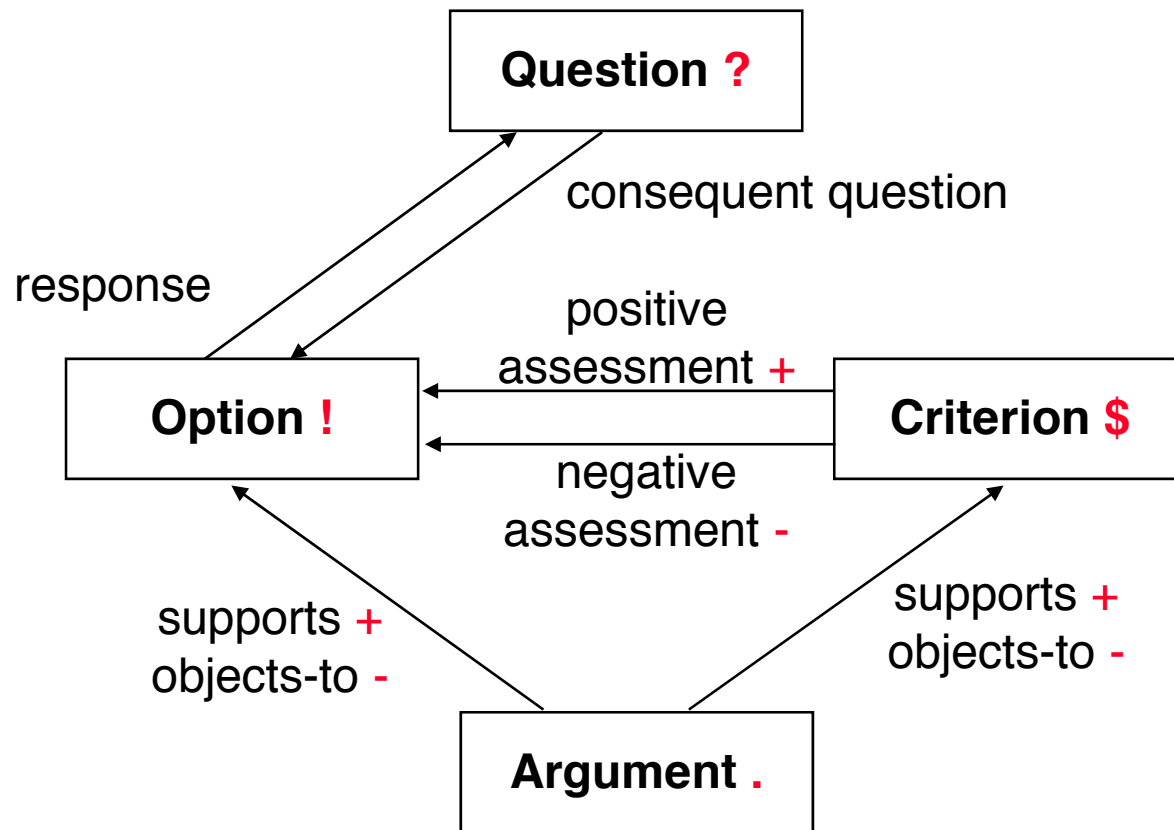
Resolutions (2)



Alternative *Issue Based Information System* Questions, Options, Criteria



- Designed for capturing rationale after the fact (e.g., quality assessment)
- QOC emphasizes criteria
- Design space explored by the developers



- What is rationale?
- Why is it critical in software engineering?
- Centralized traffic control example
- Rationale in project management
 - ◆ Consensus building (WinWin)
 - ◆ Consistency with goals (NFR Framework)
- Summary

Problem

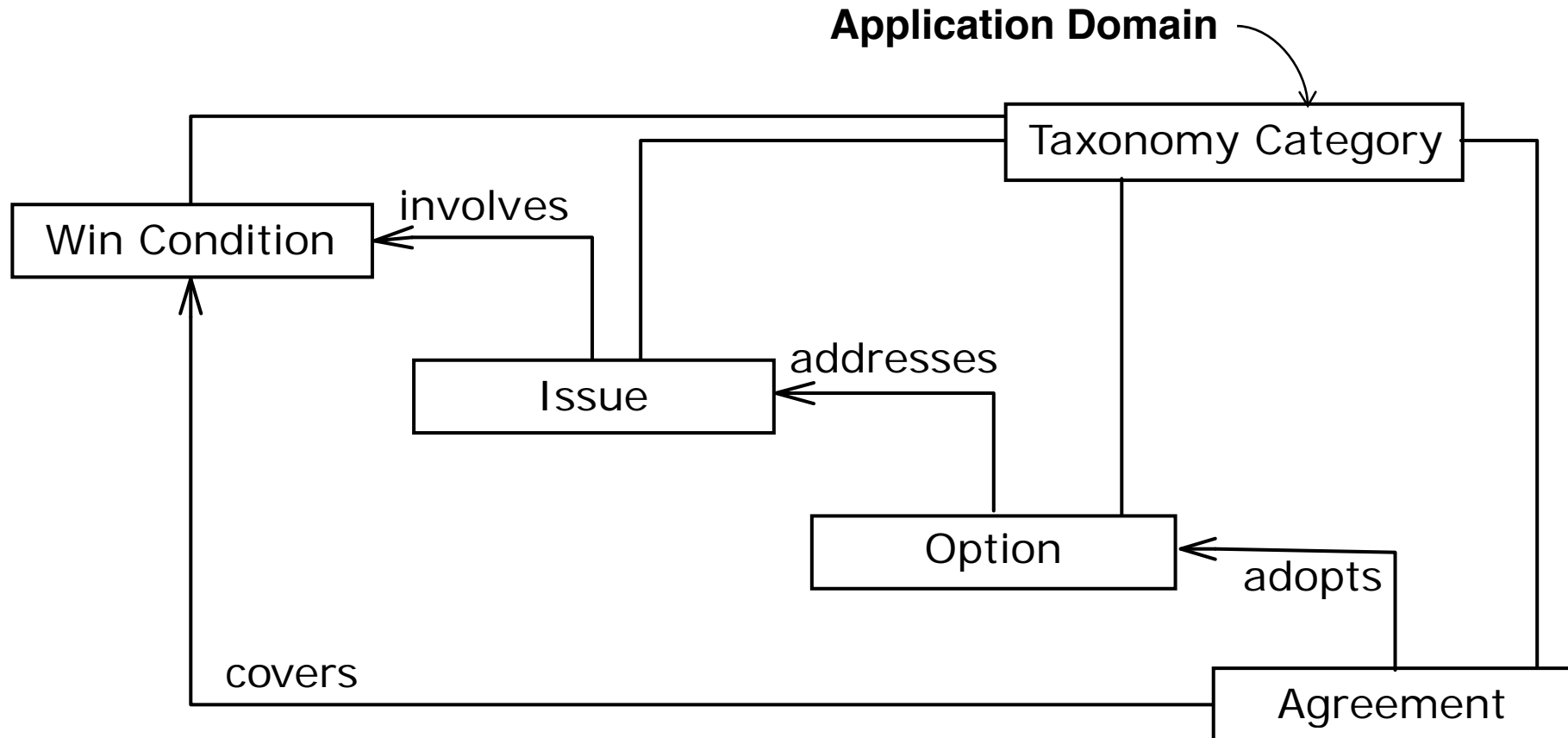
- Any realistic project suffers the tension of conflicting goals
 - ◆ Stakeholders come from different background
 - ◆ Stakeholders have different criteria

Example

- Requirements engineering
 - ◆ Client: business process (cost and schedule)
 - ◆ User: functionality
 - ◆ Developer: architecture
 - ◆ Manager: development process (cost and schedule)

- Boehm et al. '98
- Incremental, risk-driven spiral process
 - ◆ Identification of key stakeholders
 - ◆ Identification of win conditions (success criteria)
 - ◆ Conflict resolution (agreement/decision)
- Asynchronous groupware tool
 - ◆ Stakeholders post win conditions
 - ◆ Facilitator detects conflict
 - ◆ Stakeholders discuss alternatives
 - ◆ Stakeholders make agreements

Consensus building: WinWin Model



Context

- Initial case studies used project courses with real customers
- Used in industry

Results

- + Risk management focus
- + Trust building between developers and clients
- + Discipline
- Inadequate tool support

Problem

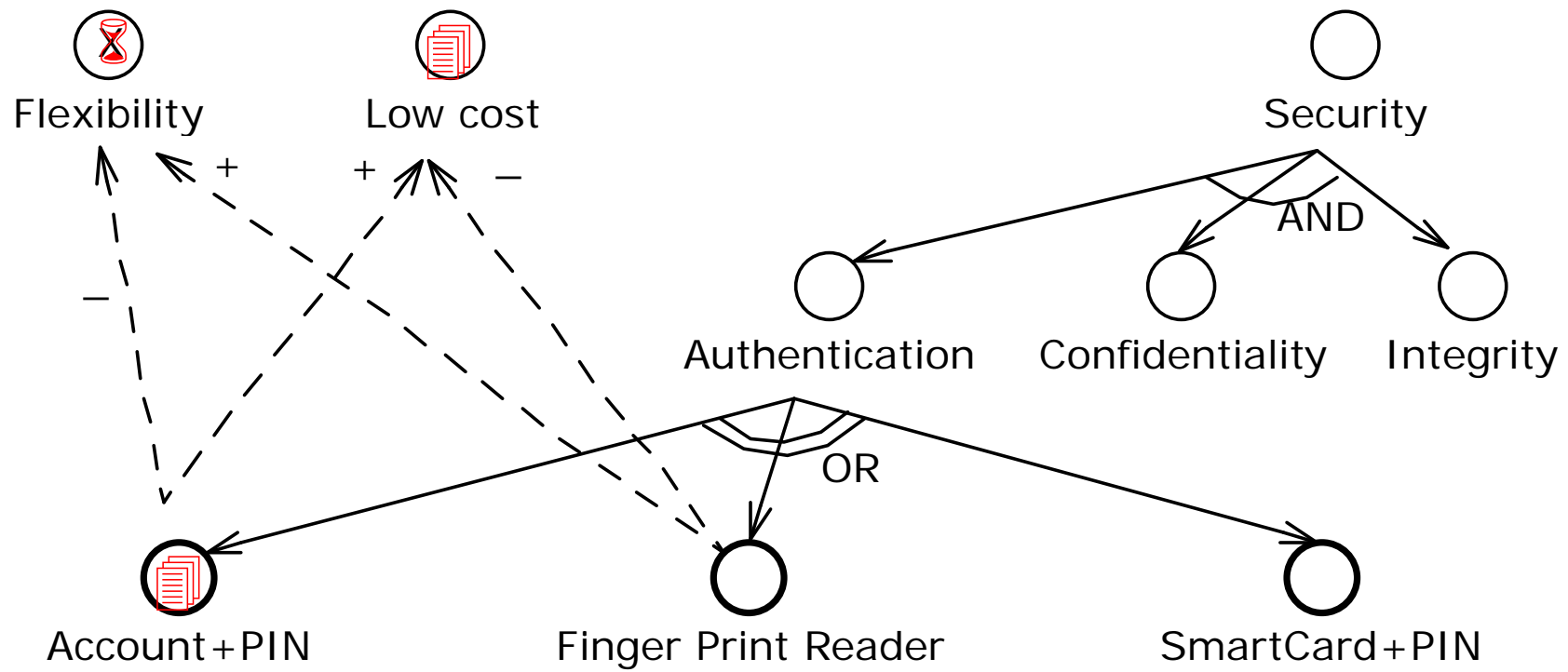
- Once multiple criteria have been acknowledged
 - ◆ Find solutions that satisfy all of them
 - ◆ Document the trade-offs that were made

Example

- Authentication should be **secure**, **flexible** for the user, and **low cost**.

- Chung et al. '99
- NFR goal refinement
 - ◆ NFRs are represented as goals in a graph
 - ◆ Leaf nodes of the graph are operational requirements
 - ◆ Relationships represent “help” “hurt” relationships
 - ◆ One graph can represent many alternatives
- NFR evaluation
 - ◆ Make and break values are propagated through the graph automatically
 - ◆ Developer can evaluate different alternatives and compare them

Consistency with goals: Model



Consistency with goals: Experiences

- + Case studies on existing systems lead to clearer trade-offs
- + Research into integrating NFR framework and design patterns
 - ◆ Match NFRs to design pattern “Forces”
 - ◆ Link NFRs, design patterns, and functional requirements
- Tool support inexistent

- Rationale can be used in project management
 - ◆ To build consensus (WinWin)
 - ◆ To ensure quality (NFR Framework)

- Other applications include
 - ◆ Risk management
 - ◆ Change management
 - ◆ Process improvement