

Lecture Notes on **Rationale Capture**

Bill Scherlis / Bernd Brügge
15-413 *Software Engineering* Fall 1999

2, 4 November 1999

Rationale Capture

- Engineering decisions
 - Requirements analysis
 - System design
 - Object design

What

Why

- Engineering decisions
 - Currency of risk
 - Safety
 - Design
 - Requirements definition
 - Technology environment
 - Build/buy
 - Frameworks
 - Team experience/competency
 - Headroom for future
 - Product line compatibility

Rationale = Justification of decisions

The Rationale for Rationale

Why preserve rationale

- System evolution
 - **The next person**
 - **Interlinked decisions**
- Environment evolution
 - **Constraints**
 - **Response to change**
- System validation
 - **Assurance beyond testing**

Why *not* preserve rationale

- Cost of representation
 - **How to express?**
 - **How to maintain consistency as system evolves?**
- Cost of capture
 - **How to elicit input?**
 - **When to elicit input?**
- Cost of access
 - **How to locate?**
 - **How to assure consistency?**
- Cost of explicitness
 - **Scrutiny**

What is Rationale?

- Rationale as motivation behind a decision
 - **Persuasion**
 - **Argumentation**

- Wicked problems (Rittel)
 - **No clear stopping rules**
 - **No objective measures of success**
 - **Iteration needed**
 - **Uncertainty of appropriate level of abstraction**
 - **Difficult to define for all stakeholders**
 - **Good reasons not to fail**
 - ***Resist top-down analysis***
 - **But design methodologies encourage top-down approaches**

What is Rationale?

- How to express and organize?

Appeal to linguistic theories of argumentation. Examples:

– ***Legal Argumentation*** (Toulmin 1958)

- *Datum* (fact)
- *Warrant* (rule)
- *Backing* (evidence)
- *Rebuttal* (exceptions)
- *Claim* (assertion)
- *Since*
- *On account of*
- *Unless*
- *So*

– ***Questions, Options, and Criteria*** (MacLean, et al. 1991)

- Question
- Option
- Argument
- Criterion

Rationale Concepts

1. Issue

- An **open** question to be addressed
 - *How to represent dates?*

2. Alternative / Proposal

- A possible solution
 - *8-bit bytes*

3. Argument

- **Criteria**
 - Desired qualities of a solution
- Justification, pro and con, with respect to criteria
 - *Compatibility with other date representations*
 - *But none of them are compatible with each other!*

4. Decision / Resolution

- Resolution of an issue, **closing** that issue
 - *8-bit bytes, with zero as 1900.*

1. Issues

- Divide and conquer
 - **Subissues**
 - *What GUI framework should we adopt?*
 - *What look-and-feel style should we adopt?*
- Coupled issues
 - **Consequence issues**
 - *How to display information to the user?*
 - *How to get information from the user?*

2. Proposals

- Can address multiple issues
 - *Select the Swing framework*
- Multiple proposals can be offered to close an issue
- Does not include argumentation

3. *Criteria and Arguments*

- Criteria can define evaluation dimensions
 - *Responsiveness*
- Criteria can define desired qualities
 - *Available*
 - *Usable*
- Argument can address a proposal, criterion, or argument

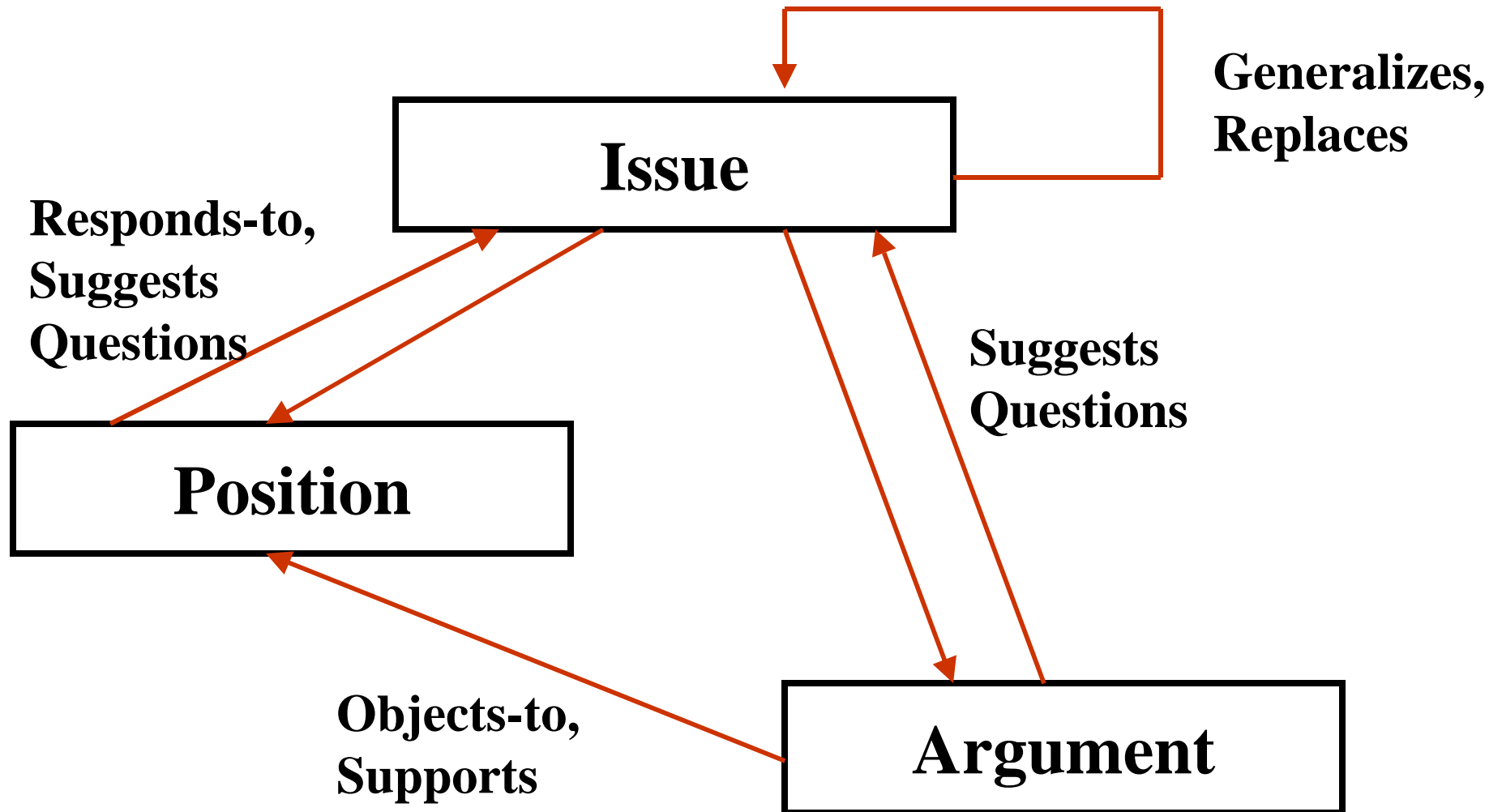
4. Resolution and Action

- Resolution is a selected alternative
- Resolution closes an issue with a decision
- Action can be the result of resolution
 - **Not part of rationale**

Representing Rationale and Argumentation

- All this can be represented in UML
- Some realizations
 - **IBIS (Issue-Based Information System, 1970)**
 - **DRL (Decision Representation Language, 1990)**
 - **QOC (Questions, Options, Criteria, 1991)**

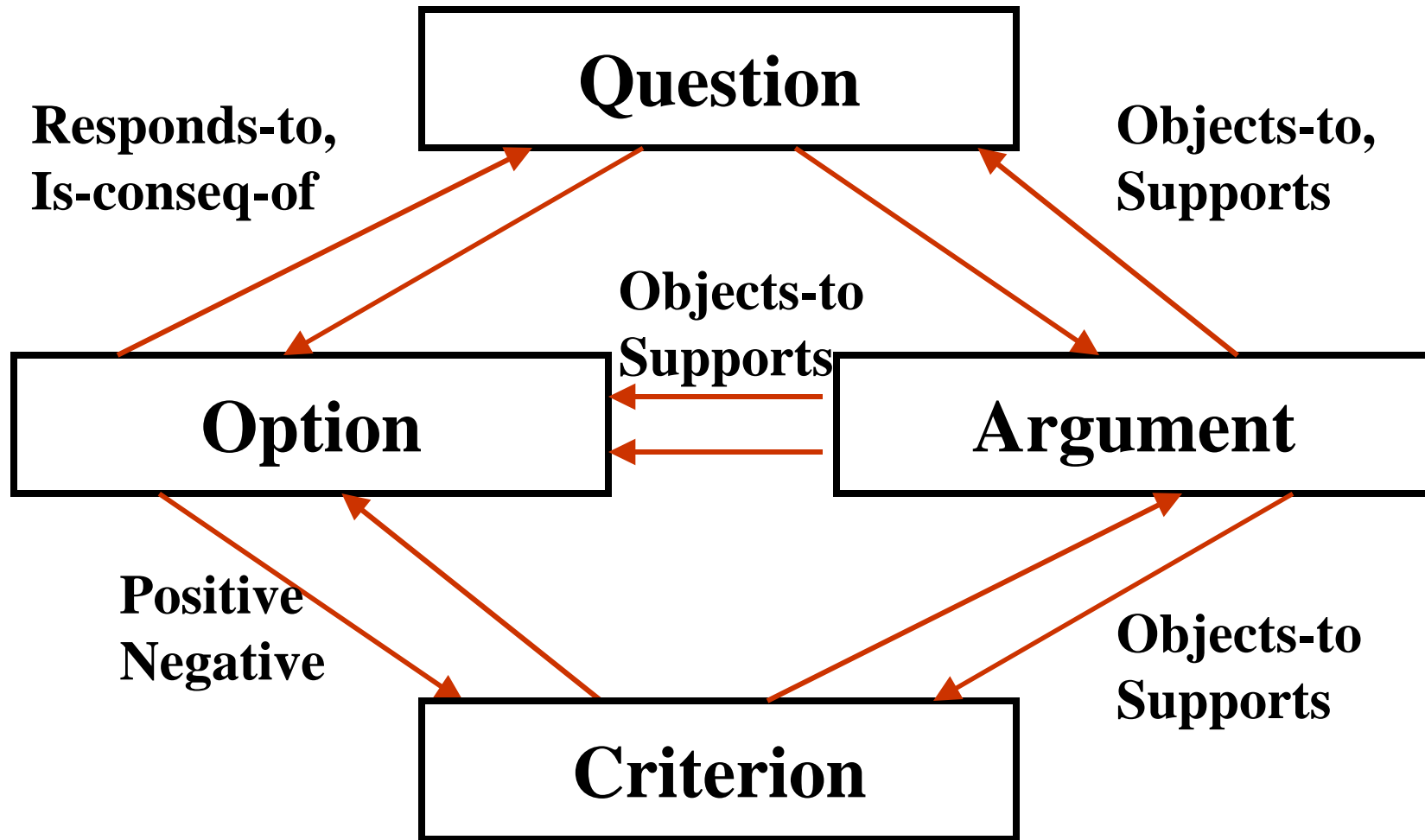
IBIS



DRL

- 7 node types
- 15 link types

QOC



Process and Product

- Capture as you go
 - “Typical” for gIBIS
 - **Deliberation is an organized *process*.**
 - *Capture during design meetings*
 - **Can constrain meeting interaction**
 - *Capture during revision*
- Capture through reflection
- “Typical” for QOC
 - **Deliberation has an organized *product*.**
 - *Revise and clarify captured rationale*
 - *Reconstruct (uncaptured) rationale*
 - **Can miss discarded alternatives**
 - **Supports review**

Decision debate

- Notes BB:
 - **Ordered by thread**
 - **Ordered by timestamp**
- Meeting:
 - **Stick to the agenda?**
 - **Chronological minutes VS. Structured minutes**
 - **I R**
 - **I P1, P2, P3 R**
 - **I P1, P2 A2.1 for, A2.2 against R**
 - **Hybridization**
 - **Type/UID labeling of each contribution**

Who Captures Rationale

Who

- Minute taker
- Rationale editor
 - **Identify, index issues**
- Reviewer

Communication

- Name the issues
- Cross-reference
 - **Make dependencies explicit**
- Manage change

When

- Incentives?
- Appropriate developer interaction
- Issues vs. resolution

Achieving consensus

- Avoid ego identification
- Criteria are the focus
- Conjoin criteria
- Negotiate
 - **Seek: expertise, management**
 - **Not: voting, owner, time**

Change

- System and Rationale must *both* change
- Recover rationale for appropriate issues
- Rework rationale as needed
 - **Exploit dependency information**

Managing Rationale

- Hypertext
- SDRD (System Design Rationale Document)
 - **Introduction**
 - **Rationale for precedent**
 - **Rationale for proposed**
 - **[SDD areas]**