AARHUS UNIVERSITET

Architectural Knowledge

Literature

[Kruchten et al., 2006]

- Kruchten, P., Lago, P., and Vliet, H.v. (2006). Building Up and Reasoning About Architectural Knowledge. In *Proceedings of QoSA 2006*

[Tyree & Akerman, 2005]

- Tyree, J. and Akerman, A. (2005). Architecture decisions: demystifying architecture. *IEEE Software, 22(2), pp 19-276*

Motivation

AARHUS UNIVERSITET

Given an architecture…

- Why is it the way it is? What is the rationale?
- What are the options as a developer? What is fixed and what is not fixed?
- What has changed since the last version?
- Can a given part of the architecture be changed?

This is not explicated with current architectural models

Architectural Knowledge

Architectural knowledge = architectural decisions + architectural design

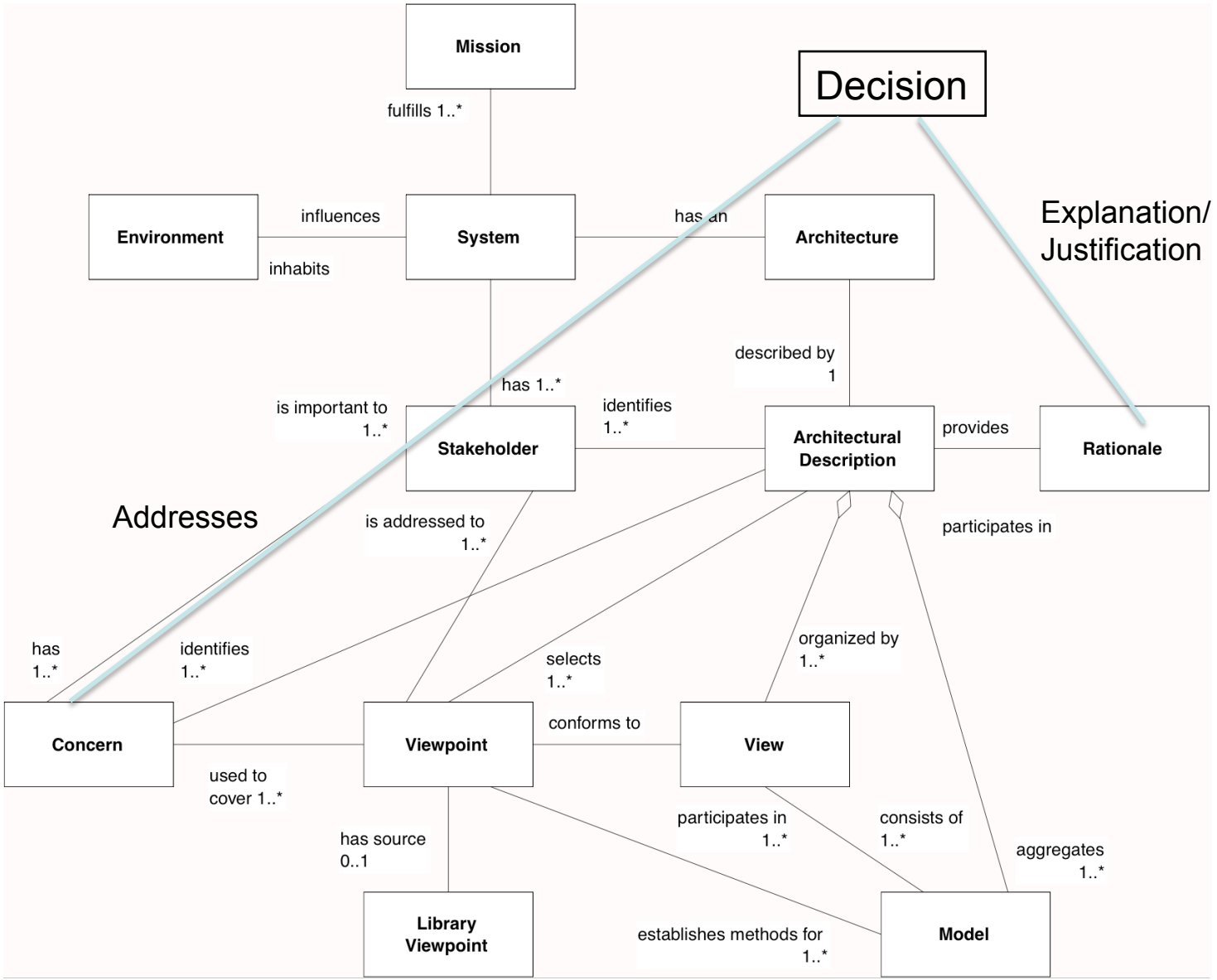
Main idea

- Making (architectural) knowledge explicit aids in producing quality systems

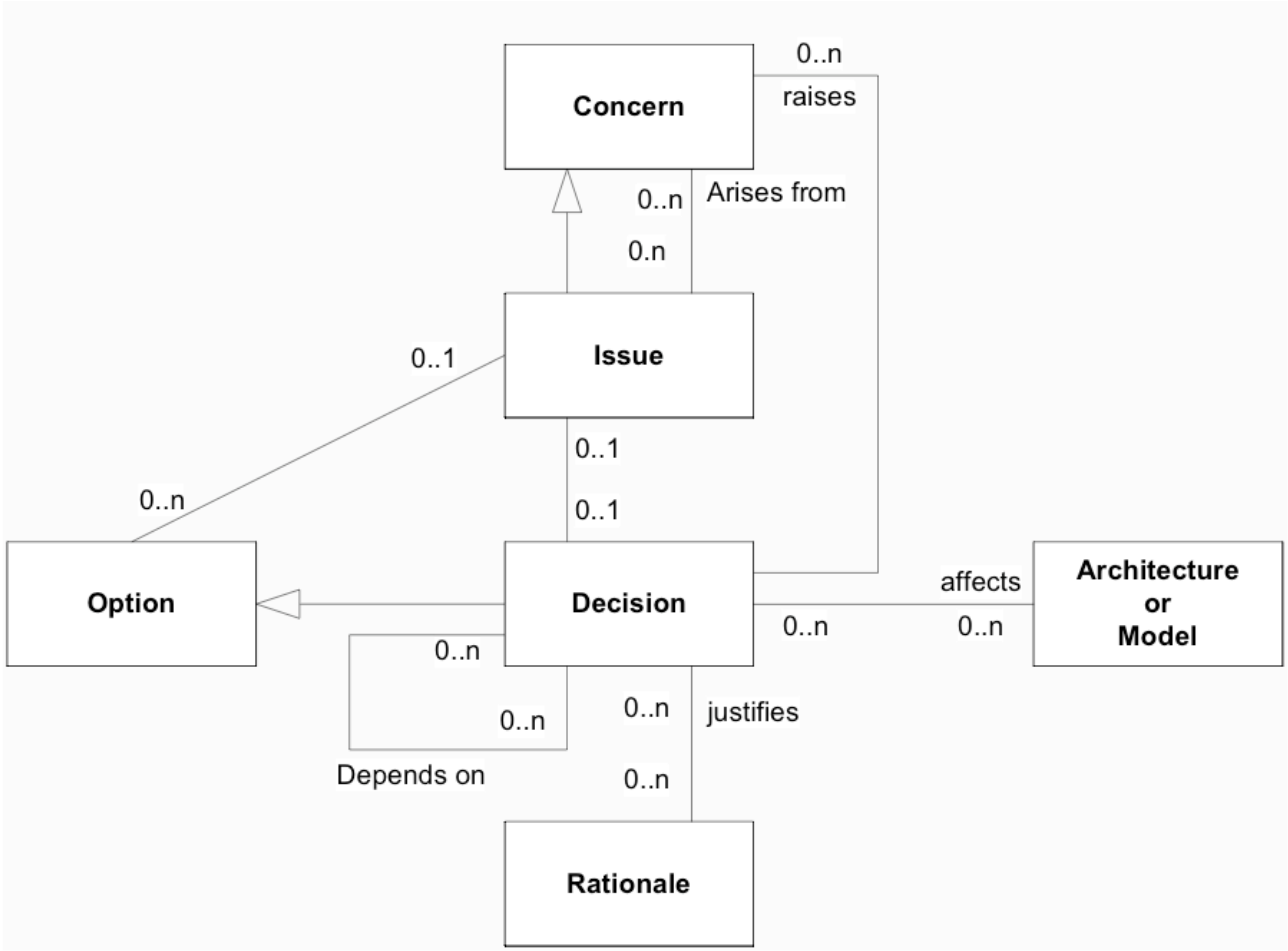
Issues

- What are “architectural decisions”?
- How to explicate architectural decisions (efficiently)?

IEEE 1471

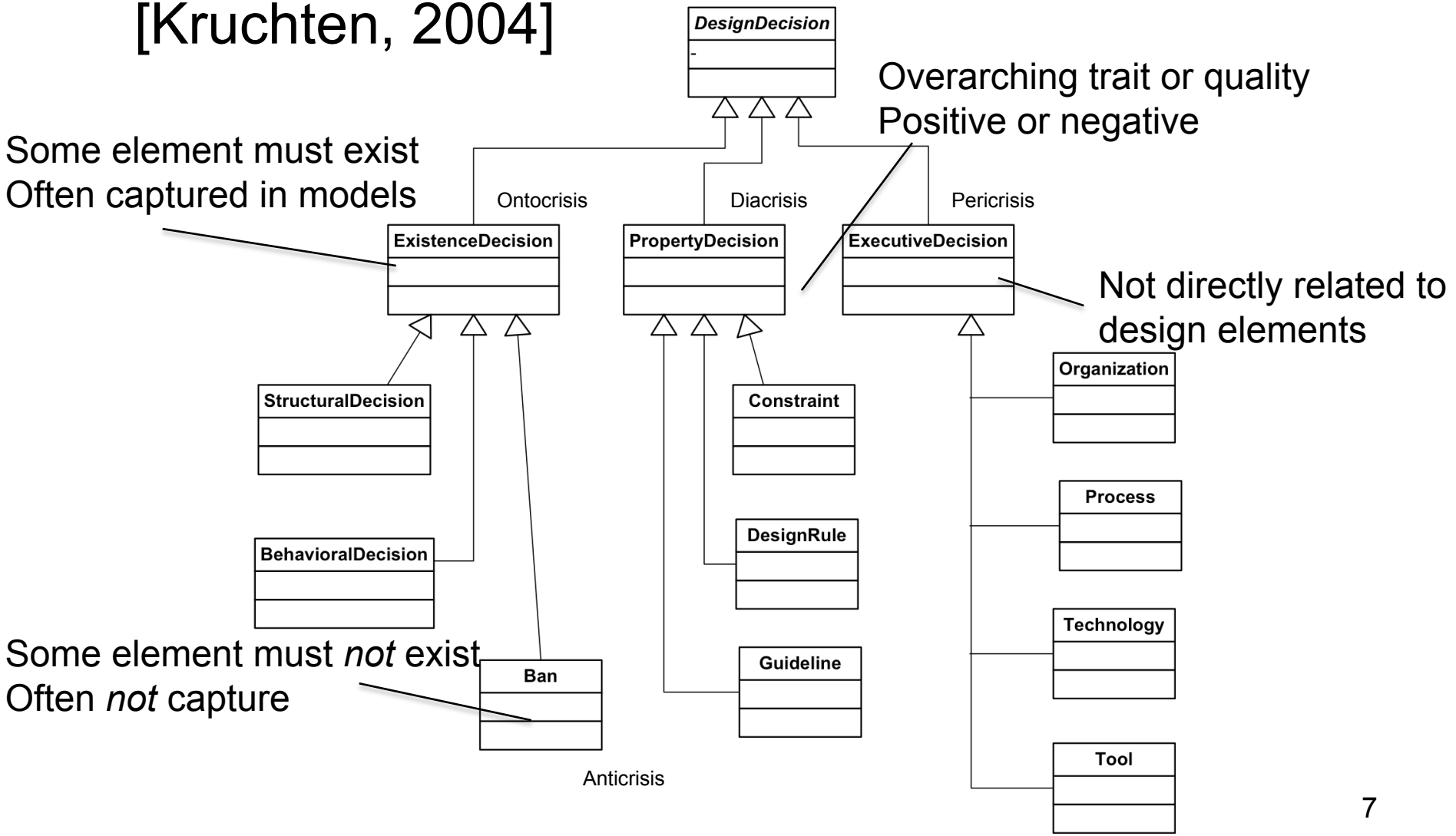

Mission
Decision
fulfills 1..*
Environment
influences
inhabits
System
has an
Architecture
Explanation/
Justification
described by
1
has 1..*
is important to
1..*
identifies
1..*
Stakeholder
Architectural
Description
provides
Rationale
Addresses
is addressed to
1..*
participates in
has
1..*
identifies
1..*
selects
1..*
organized by
1..*
Concern
Viewpoint
conforms to
View
used to
cover 1..*
participates in
1..*
consists of
1..*
has source
0..1
aggregates
1..*
Library
Viewpoint
establishes methods for
1..*
Model

[Avgeriou et al., 2007]



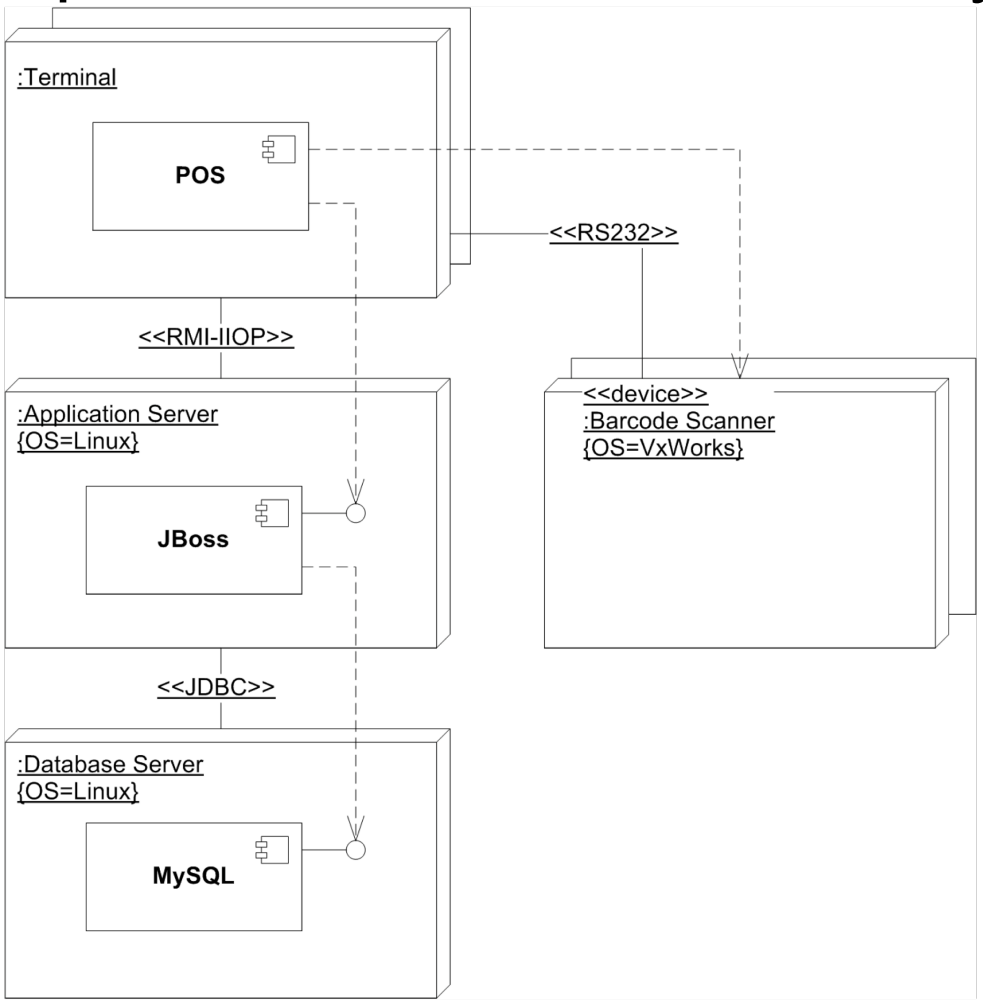
Types of Decisions

[Kruchten, 2004]



Exercise

List possible decisions of each type for POS



What should be documented?

Attributes of design decisions [Kruchten et al., 2006]

- Epitome (the decision itself)
- Rationale
- Scope
  - System, time, organization scope
- Author, timestamp, history
- State
  - Idea, tentative, decided, approved, rejected, challenged, obsolete
- Categories
  - Keyword based on, e.g., quality, concern, ontology
- Relations among decisions
  - Constrains, forbids, excludes, enables, subsumes, conflicts with, overrides, comprises, is an alternative to, is bound to, is related to, dependency
- Relations to other artifacts

Exercise

AARHUS UNIVERSITET

Take a decision from previous exercise and describe it using this template

What should be documented?

[Tyree and Akerman, 2005]

Issue	Describe the architectural design issue you're addressing, leaving no questions about why you're addressing this issue now. Following a minimalist approach, address and document only the issues that need addressing at various points in the life cycle.
Decision	Clearly state the architecture's direction—that is, the position you've selected.
Status	The decision's status, such as pending, decided, or approved.
Group	You can use a simple grouping—such as integration, presentation, data, and so on—to help organize the set of decisions. You could also use a more sophisticated architecture ontology, such as John Kyaruzi and Jan van Katwijk's, which includes more abstract categories such as event, calendar, and location.[8] For example, using this ontology, you'd group decisions that deal with occurrences where the system requires information under event.
Assumptions	Clearly describe the underlying assumptions in the environment in which you're making the decision—cost, schedule, technology, and so on. Note that environmental constraints (such as accepted technology standards, enterprise architecture, commonly employed patterns, and so on) might limit the alternatives you consider.
Constraints	Capture any additional constraints to the environment that the chosen alternative (the decision) might pose.
Positions	List the positions (viable options or alternatives) you considered. These often require long explanations, sometimes even models and diagrams. This isn't an exhaustive list. However, you don't want to hear the question "Did you think about … ?" during a final review; this leads to loss of credibility and questioning of other architectural decisions. This section also helps ensure that you heard others' opinions; explicitly stating other opinions helps enroll their advocates in your decision.
Argument	Outline why you selected a position, including items such as implementation cost, total ownership cost, time to market, and required development resources' availability. This is probably as important as the decision itself.
Implications	A decision comes with many implications, as the REMAP metamodel denotes. For example, a decision might introduce a need to make other decisions, create new requirements, or modify existing requirements; pose additional constraints to the environment; require renegotiating scope or schedule with customers; or require additional staff training. Clearly understanding and stating your decision's implications can be very effective in gaining buy-in and creating a roadmap for architecture execution.
Related decisions	It's obvious that many decisions are related; you can list them here. However, we've found that in practice, a traceability matrix, decision trees, or metamodels are more useful. Metamodels are useful for showing complex relationships diagrammatically (such as Rose models).
Related requirements	Decisions should be business driven. To show accountability, explicitly map your decisions to the objectives or requirements. You can enumerate these related requirements here, but we've found it more convenient to reference a traceability matrix. You can assess each architecture decision's contribution to meeting each requirement, and then assess how well the requirement is met across all decisions. If a decision doesn't contribute to meeting a requirement, don't make that decision.
Related artifacts	List the related architecture, design, or scope documents that this decision impacts.
Related principles	If the enterprise has an agreed-upon set of principles, make sure the decision is consistent with one or more of them. This helps ensure alignment along domains or systems.
Notes	Because the decision-making process can take weeks, we've found it useful to capture notes and issues that the team discusses during the socialization process.

Exercise

AARHUS UNIVERSITET

Take a decision from previous exercise and describe it using this template

Barriers to Adoption

AARHUS UNIVERSITET

Overhead in capturing design decisions

- We will explore relation to backlog in H5
- May tailor decision templates

Capturing is not enough

- *Use case model*

Use Case Model – Actors

Users of architectural knowledge

- Architects (designing the system)
- Developers
- Reviewers
- Analysts
- Maintainers
- Re-users
- Software tools

Use Case Model – Use Cases

Uses of architectural knowledge

- Review
  - Incremental review
  - Review for a specific concern
  - Evaluate impact
- Get a rationale
- Study chronology
- Add a decision
- Stakeholder
  - Spot the subversive stakeholder
  - Spot the critical stakeholder
- Clone architectural knowledge
- Integration
- Detection of patterns

Summary

AARHUS UNIVERSITET

Architecture is more than architecture models

- A lot of decisions are tacit
  - Existence, ban, property, executive decisions
- Could benefit from making decisions explicit (even formalized)

Architectural knowledge

- Architectural design
- Architectural decisions

Both tool support and theory are not mature yet