Introduction to the Techne family of formalisms and notes on the design of formalisms for early-stage Requirements Engineering

Ivan Jureta
FNRS & Louvain School of Management,
University of Namur
Belgium
ivan.jureta@fundp.ac.be

Content is based on past and ongoing work with Alexander Borgida, John Mylopoulos and Neil Ernst.

Topic & Overview

- Change of requirements is reflected in changes of representations of requirements

- Talk focuses on formalisms for representation and reasoning about requirements

- Understanding content of representations should help understand requirements change and consequences in terms of system adaptation

Talk in three parts:

1. Why Techne?
2. What is Techne?
3. What we learned on the way?

Starting point

Zave & Jackson

"Four dark corners of requirements engineering"

ACM TOSEM 6(1), 1997

Synthetic requirements problem statement:

Given R and K, find S such that $\mathbf{K}, \mathbf{S} \vdash \mathbf{R}$

Teaches the following key ideas:

1.	**Types**	There are R, K and S formulas
2.	**Consistency**	If R is consistent, K and S must be consistent
3.	**Achievement**	Deduce every R formula from K and S

Designing a formalism for the Z&J requirements problem

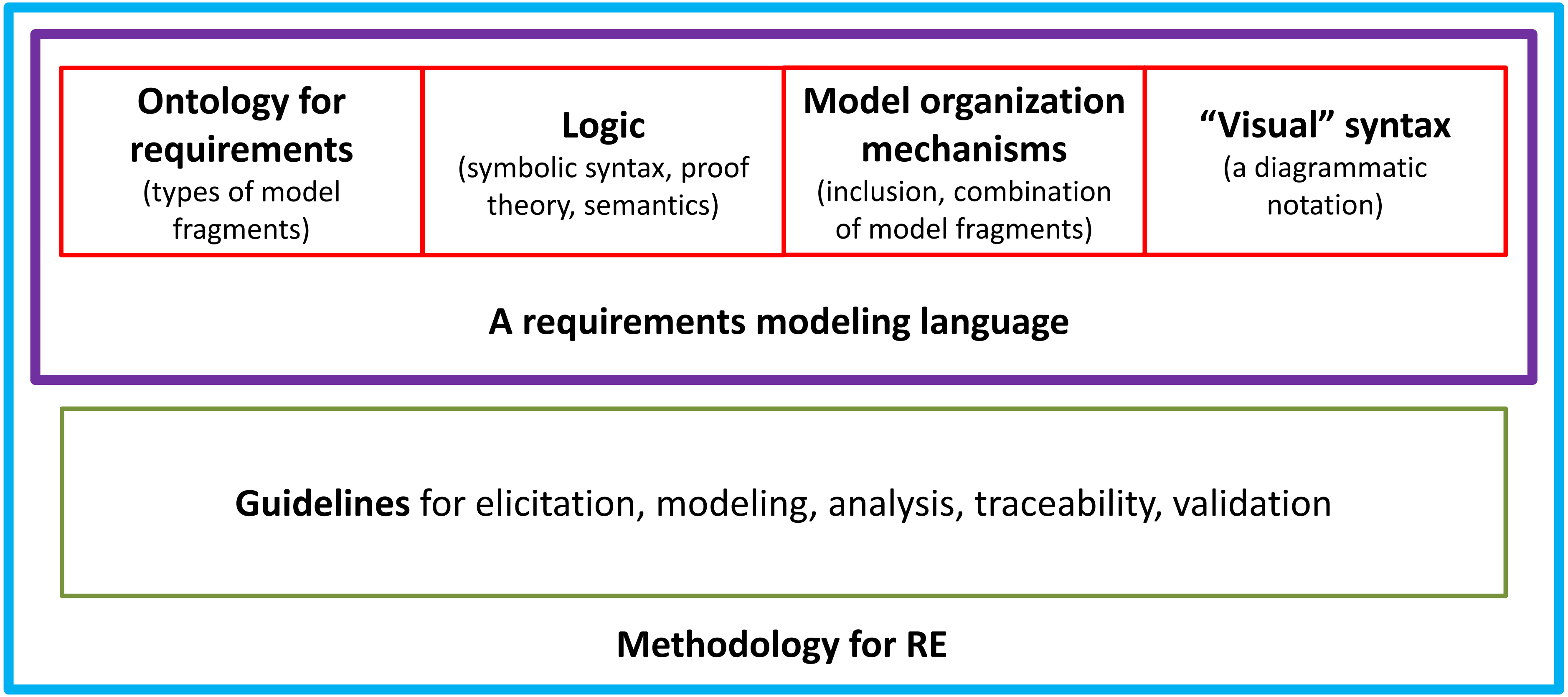
Ingredients:

1. Three types
    1. R: Requirements (what is desired)
    2. K: Domain assumptions (what is true)
    3. S: Specification (what to do)
2. A logic

Rough steps to make a formalism for representation and reasoning in Requirements Engineering (RE)

1. Choose types
2. Choose a logic
3. Write methodology on how to convert requirements statements into typed formulas

Formalisms for RE = Requirements modeling languages (RMLs)



Formalisms for RE = Requirements modeling languages (RMLs)

Examples:

- **“Original” RML**

  Greenspan, Borgida, Mylopoulos. *Info. Sys.*, 11(1), 1986.
- **ERAE**

  Dubois, Hagelstein, Rifaut. *Philips J. of Res.*, 43(3-4), 1988.
- **KAOS**

  Dardenne, van Lamsweerde, Fickas. *Sci. Comp. Prog.*, 20(1-2), 1993.
- **i-star**

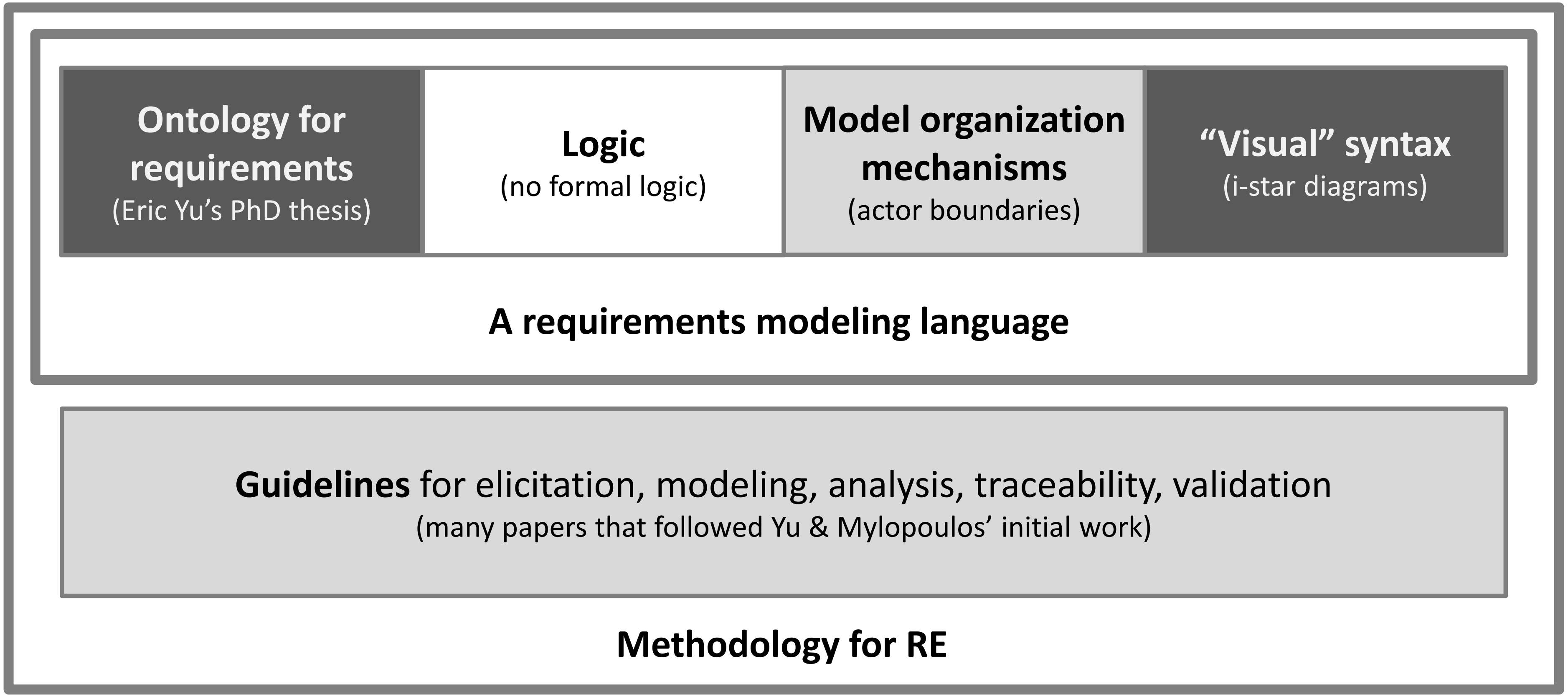
  Yu, Mylopoulos. *ICSE*, 1994.
- **Tropos**

  Castro, Kolp, Mylopoulos. *Info. Sys.*, 27(6), 2002.
- **Formal Tropos**

  Fuxman, Liu, Mylopoulos, Roveri, Traverso. *Req. Eng.*, 9(2), 2004.

Formalisms for RE = Requirements modeling languages (RMLs)

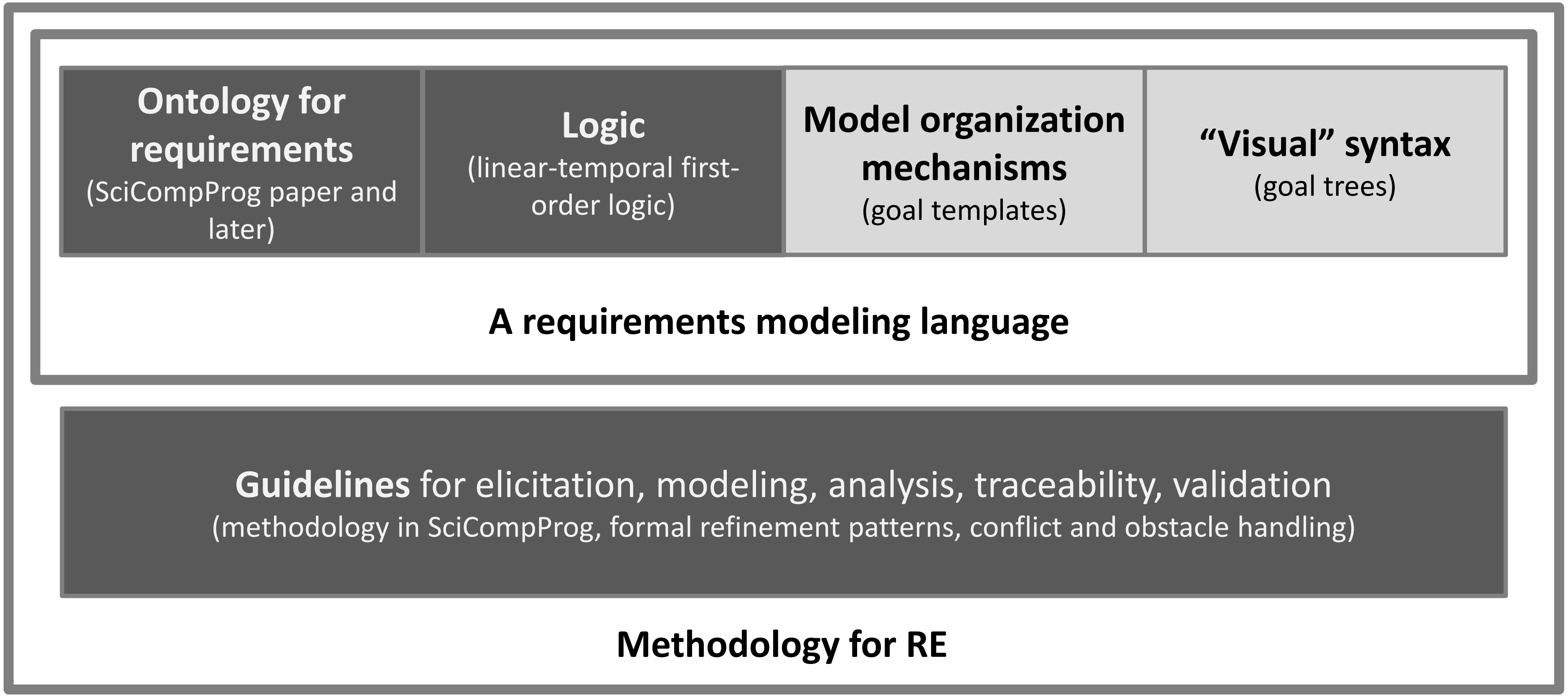
Quick overview: Components of **i-star**



Color intensity: the darker the color, the more the formalism seems to be advanced in the colored component

Formalisms for RE = Requirements modeling languages (RMLs)

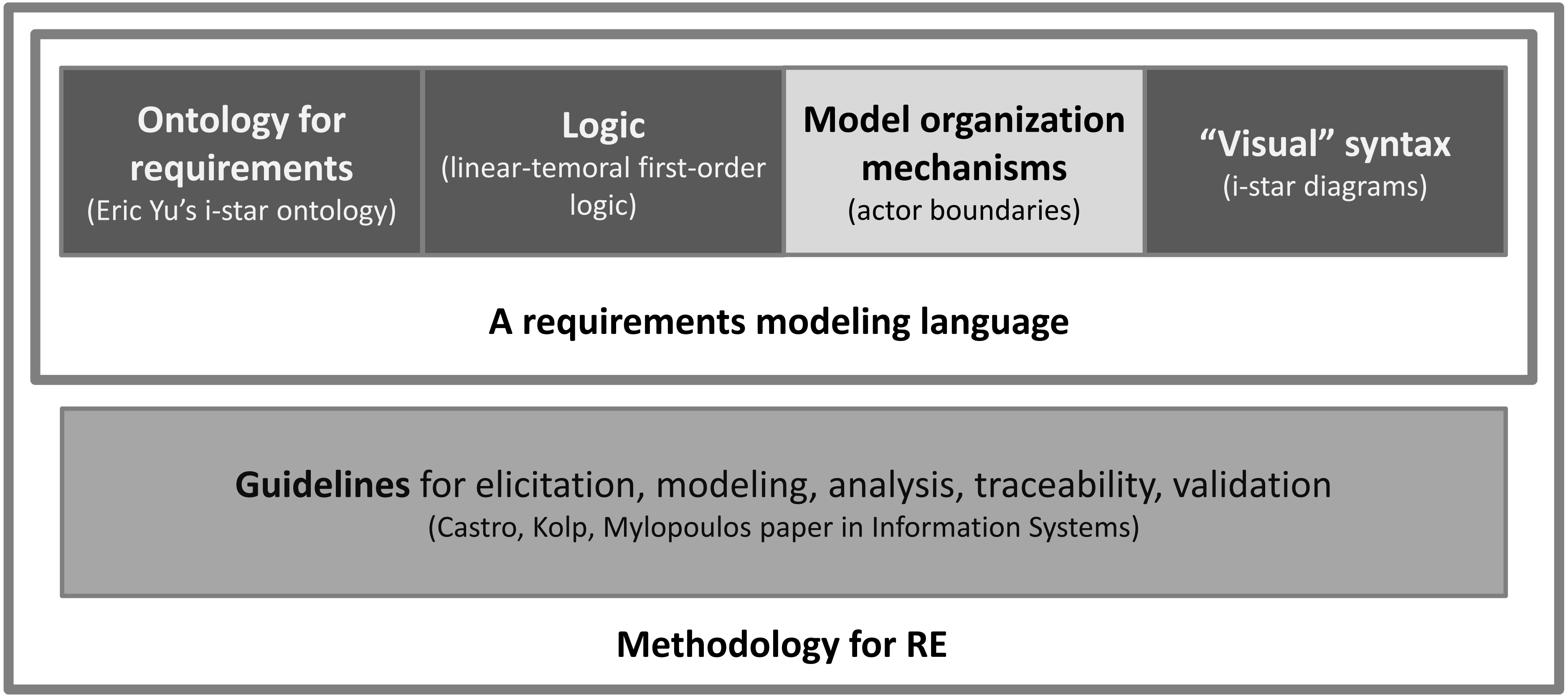
Quick overview: Components of **KAOS**



Color intensity: the darker the color, the more the formalism seems to be advanced in the colored component

Formalisms for RE = Requirements modeling languages (RMLs)

Quick overview: Components of **Formal Tropos**



Color intensity: the darker the color, the more the formalism seems to be advanced in the colored component

Approach in Techne

We want to design all components according to knowledge specific to RE.

This means the structure we want is less modular. It looks more like this:

Guidelines for elicitation, modeling, analysis, traceability, validation

Ontology for requirements (types of model fragments)

Logic (symbolic syntax, proof theory, semantics)

Model organization mechanisms (inclusion, combination of model fragments)

"Visual" syntax (a diagrammatic notation)

Rather than:

Ontology for requirements (types of model fragments)	**Logic** (symbolic syntax, proof theory, semantics)	**Model organization mechanisms** (inclusion, combination of model fragments)	**"Visual" syntax** (a diagrammatic notation)
Guidelines for elicitation, modeling, analysis, traceability, validation			

So we first went back to the requirements problem

And we asked the following questions:

1. Is there always one problem statement?
2. Is there always one solution to a problem statement?
3. If not, does the statement say anything about the comparison of solutions?
4. Does the statement say anything about the quality of a solution?
5. Where are requirements which can only be partially satisfied?
6. Do we have to satisfy all requirements in R and maintain all domain assumptions in K?
7. If not, how do we choose which R and K to drop, and which to keep?
8. Is R always consistent?
9. Are K and S always consistent?
10. If not, what does the problem statement say when K, S, or R are inconsistent?

Which led to the following **restatement of the problem (RE2008)**

Given the elicited

domain assumptions

goals

quality constraints

softgoals

tasks

Find

tasks & domain assumptions which

satisfy all mandatory goals and quality constraints,

and if feasible, satisfy many preferred requirements

and many optional requirements.

Which in turn led to **the original Techne (RE2010)**

Overall aim

To make the simplest RML which would:

1. Include all types of formulas from the restated requirements problem
2. Allow the definition of the restated problem and solution concepts

Types in Techne:

- **g**: Goal
- **s**: Softgoal
- **q**: Quality constraint
- **t**: Task
- **k**: Domain assumption
    - **Refinement** formulas
    - **Operationalization** formulas
    - **Conflict** formulas

Syntax:

$$pl ::= \mathbf{k}(p) \mid \mathbf{g}(p) \mid \mathbf{q}(p) \mid \mathbf{s}(p) \mid \mathbf{t}(p)$$

$$\phi ::= \bigwedge_{i=1}^{n} pl_i \to pl \mid \bigwedge_{i=1}^{n} pl_i \to \bot$$

$$sn ::= pl \mid \mathbf{k}(\phi)$$

Consequence relation $\mid\!\sim_{\mathcal{T}}$

Reflexive and paraconsistent **consequence relation** which lets rule consequents through when antecedents can be deduced.

And these **problem and solution concepts**

Problem: **Given G, S, Q, K, find the solution to the requirements problem.**

Candidate solution is a set of tasks T* and domain assumptions K* s.t.:

1. $K^*, T^* \not\vdash_{\mathcal{T}} \perp$
2. $K^*, T^* \vdash_{\mathcal{T}} G^*, Q^*, \text{ where } G^* \subseteq G, \; Q^* \subseteq Q$
3. G* and Q* include, respectively, all mandatory goals and quality constraints
4. All mandatory softgoals are approximated by the consequences of K* and T*, that is: $K^*, T^* \not\vdash_{\mathcal{T}} S^M$

Solution is the candidate solution which ranks highest according to a **decision rule**.

Decision rule is a procedure by which a ranking is obtained from preference relations between Techne formulas.

Lessons learned

- It is hard to design a formalism starting from methodological intuitions, but it is a relevant investment because it requires one to clarify considerably these methodological intuitions.
- It is also a different effort than designing a methodology on top of an existing and well-known formalism.
- Types can complicate a formalism considerably when they are not used purely for the organization of formulas, but have a role in the proof theory.

Limitations of the original Techne:

- It oversimplifies the Softgoal and Quality constraint types.
- It handles inconsistency in a much too simple way.
- It has no mechanisms to organize a model.
- It lacks interesting types.
- And so on.

…led us to formulate guidelines we are now following

1. Techne is not one formalism, but a family of formalisms.
2. Each Techne formalism is:
   - A propositional formalism, to preserve natural language statements of requirements.
   - Constrained to some specific set of types.
   - Comes with its own problem and solution concepts.
3. RE involves both problem-solving and decision-making, and both need to be reflected in Techne formalisms, so the formalism needs
   - ability to compare formulas according to different criteria
   - ability to aggregate comparisons to produce a ranking of solutions
4. When a set is inconsistent depends on what role inconsistency has in a formalism.
5. When a new type comes up, it is added to one (the simplest possible) Techne formalism by following specific rules, which helps study these types in a better controlled (simpler) environment.

Ongoing work: Techne family in 2011

Table 1: Overview of the Techne formalisms according to their components; Symbol "•" reads "present", "○" reads "absent".

Components		Formalisms T1 (§5)	T2 (§6)	T3 (§7)	T4 (§8)	T5 (§9)
Core modalities	Goal	•	•	•	•	•
	Domain assumption	•	•	•	•	•
	Task	•	•	•	•	•
	Quality constraint	○	○	○	○	•
	Softgoal	○	○	•	•	•
	Soft dom. assumpt.	○	○	•	•	•
Optionality modalities	Mandatory	○	○	○	•	•
	Preferred	○	○	○	•	•
	Inherited	○	○	○	•	•
Agency modalities	Agent	○	•	•	•	•
	Role	○	•	•	•	•
Primitive relations	Inference	•	•	•	•	•
	Conflict	•	•	•	•	•
	Responsibility	○	•	•	•	•
	Ability	○	•	•	•	•
	Occupancy	○	•	•	•	•
	Commitment	○	•	•	•	•
	Preference	○	○	•	•	•
Variables	Binary Boolean	•	•	•	•	•
	Rational	○	○	○	○	•

Ongoing work: Types we are exploring

Psychological attitude types			Variable types		Clarity types		Resulting types:
Desired	Believed	Intended	Binary	Rational	Clear	Vague	
							Goal (G)
							Domain assumption (K)
							Task (T)
							Quality constraint (Q)
							Quantitative K
							Quantitative T
							Soft G
							Soft K
							Soft T

Ongoing work: Types we are exploring

Examples of typed propositions

- **Goal**(ambulance is dispatched to incident location)
- **Domain_assumption**(callers report imprecise incident locations)
- **Task**(use radio to communicate incident location to ambulance)
- **Quality_constr**(t < 12 minutes, t is the time between the reception of the call to the moment when the ambulance confirms arrival to incident location reported in the call)
- **Quant_dom_assumpt**(t = x + y + z, where x is the time to identify location, y is the time to identify and dispatch available ambulance, and z is time for dispatched ambulance to arrive at incident location and report arrival at the incident location)
- **Quant_task**(purchase n new ambulances per year)
- **Soft_goal** (low average ambulance response time)
- **Soft_dom_assumpt**(average number of calls is low on the first week of December)
- **Soft_task**(the number of available ambulances will be optimized)

Ongoing work: Types we are exploring

Deontic modality			Variable types		Clarity types		Resulting types:
Obligatory	Permitted	Prohibited	Binary	Rational	Clear	Vague	
■			■		■		Obligation (O)
	■		■		■		Permission (P)
		■	■		■		Prohibition (F)
■				■	■		Quantitative O
	■			■	■		Quantitative P
		■		■	■		Quantitative F
■			■			■	Soft O
	■		■			■	Soft P
		■	■			■	Soft F

Ongoing work: Types of relations we are exploring

E.g., each Techne formalism comes with its own operationalization relation:

In T1:

Goal g is operationalized by the set O iff

- O is consistent
- every member of O is a task or a domain assumption.

In T2:

Goal g is operationalized by the set O iff

- O is consistent
- every member of O is a task or a domain assumption
- there is a role which is responsible for every member of O.

Ongoing work: Properties of consequence relations

Examples of properties that we may want the consequence relation of the formalism to satisfy:

1. Not allow us to derive any formula from an inconsistent set of formulas.
2. Not block us from deducing formulas from an inconsistent set of formulas.
3. Deduce at least the same inconsistencies as some other benchmark formalism.
4. Deduce at most the same inconsistencies as some other benchmark formalism.
5. Derive inconsistency from circular refinements and operationalizations.
6. Enable inconsistency resolution without contraction.

Thank you

Backup slides

Overview

- Change of requirements is reflected in changes of representations of requirements
- Talk focuses on representations of requirements
- More specifically on formalisms for representation and reasoning about requirements
- Simple observations about requirements will be given
- Consequences of these observations will be discussed
- Based on the identified consequences, guidelines will be suggested for the design of formalisms for representation and reasoning in RE
- Examples will be given on how these guidelines translate into formalism design for Requirements Engineering (RE)

Observation 1:

Different stakeholders have different expertise.

Observation 1:

Different stakeholders have different expertise.

- A major purpose of representations is to facilitate communication.
- For whom?
  - Business analysts / requirements engineers
  - Engineers and engineering management
  - Business strategy
  - Marketing
  - Legal (compliance)
  - Other, non-engineering parties
- They have different viewpoints, in the traditional sense in RE.
- But also: they use different languages and focus on specific questions that can be answered through reasoning on representations in such languages.

Observation 1:

Different stakeholders have different expertise.

Thinking about expertise-specific RE formalisms comes with some benefits:

- It highlights the importance of studying formalisms for early-stage RE.
- It highlights the target audience of a formalism and perhaps suggests that the quality of the formalism will depend in its ability to convey information relevant to problem-solving and decision-making by that target audience.
- RE formalisms have focused significantly on software engineers. There are other groups of participants, and while they do not speak the software engineering languages, they do have relevant things to say.

Observation 1:

Different stakeholders have different expertise.

Guideline A
Propositional formalisms preserve statements of requirements in natural language. This makes them more convenient for non-engineering target audiences than first-order formalisms.

Guideline B
Types of formulas in a formalism should reflect categories relevant to the target audience.

Observation 1:

Different stakeholders have different expertise.

Applying the guidelines: choosing types for stakeholder groups

Psychological attitude types			Variable types		Clarity types		Resulting types:
Desired	Believed	Intended	Binary	Rational	Clear	Vague	
							Goal (G)
							Domain assumption (K)
							Task (T)
							Quality constraint (Q)
							Quantitative K
							Quantitative T
							Soft G
							Soft K
							Soft T

Observation 1:

Different stakeholders have different expertise.

Applying the guidelines: choosing types for stakeholder groups

Deontic modality			**Variable types**		**Clarity types**		**Resulting concept:**
Obligatory	**Permitted**	**Prohibited**	**Binary**	**Rational**	**Clear**	**Vague**	
							Obligation (O)
							Permission (P)
							Prohibition (F)
							Quantitative O
							Quantitative P
							Quantitative F
							Soft O
							Soft P
							Soft F

Observation 1:

Different stakeholders have different expertise.

Applying the guidelines: typed propositions

- **Goal**(ambulance is dispatched to incident location)
- **Domain_assumption**(callers report imprecise incident locations)
- **Task**(use radio to communicate incident location to ambulance)
- **Quality_constr**(t < 12 minutes, t is the time between the reception of the call to the moment when the ambulance confirms arrival to incident location reported in the call)
- **Quant_dom_assumpt**(t = x + y + z, where x is the time to identify location, y is the time to identify and dispatch available ambulance, and z is time for dispatched ambulance to arrive at incident location and report arrival at the incident location)
- **Quant_task**(purchase n new ambulances per year)
- **Soft_goal** (low average ambulance response time)
- **Soft_dom_assumpt**(average number of calls is low on the first week of December)
- **Soft_task**(the number of available ambulances will be optimized)

Observation 2:

Different stakeholders have different decision authority.

Observation 2:

Different stakeholders have different decision authority.

- Stakeholders can be related through formal hierarchy.

- Position in the hierarchy influences:
  - Domains in which decisions can be made
  - Ability to overrule others' decisions

- This has to do with inconsistency: Differences in positions means there will be inconsistencies which cannot be resolved as soon as they are found.

- This should not stop us from drawing meaningful conclusions from the representation of requirements.

Observation 2:

Different stakeholders have different decision authority.

Guideline C
How inconsistency should be tolerated depends on what inconsistency should convey in problem-solving and decision-making.

Guideline D
How inconsistency is tolerated needs to be decided when the formalism is being designed.

Observation 2:

Different stakeholders have different decision authority.

Applying the guidelines

Examples of properties that we may want the consistency relation of the formalism to satisfy:

1. Not allow us to derive any formula from an inconsistent set of formulas.
2. Not block us from deducing formulas from an inconsistent set of formulas.
3. Deduce at least the same inconsistencies as some other benchmark formalism.
4. Deduce at most the same inconsistencies as some other benchmark formalism.
5. Derive inconsistency from circular refinements and operationalizations.
6. Enable inconsistency resolution without contraction.

Observation 3:

RE involves both problem-solving and decision-making.

Observation 3:

RE involves both problem-solving and decision-making.

- Support for problem-solving involves at least:
  - Ability to draw conclusions from a representation of requirements
  - Relation between representations at different levels of abstraction (i.e., refinement relation)

- Support for decision-making requires:
  - Notion of the problem to solve
  - Notion of solution
  - Way to identify solutions
  - Way to compare solutions over many criteria
  - Way to aggregate criteria into a ranking

Observation 3:

RE involves both problem-solving and decision-making.

Guideline E
A formalism for RE needs to incorporate a taxonomy of problem and solution concepts.

Guideline F
A formalism for RE needs to include features allowing the comparison of solution fragments and this over various criteria.

Guideline G
A formalism for RE needs features used to aggregate comparisons of fragments of solutions, i.e., features that allow the definition of decision rules.

Observation 3:

RE involves both problem-solving and decision-making.

Applying the guidelines: Taxonomy of solution concepts influenced by formalism features

Suppose we have a formalism where each formula is one of three types:
- Goal (all goals are in set G)
- Domain assumption (set K)
- Tasks (set T)

Solution is defined as a set of formulas X* which satisfies three conditions:
1. X* is a subset of (K union T)
2. X* is consistent
3. Every member of G can be deduced from X*

Observation 3:

RE involves both problem-solving and decision-making.

Applying the guidelines: Taxonomy of solution concepts influenced by formalism features

Suppose we add the following to the formalism with goals, domains assumptions and tasks:

- Set of symbols for roles (of an actor)
- Relation between Role and typed formula (e.g., role responsible for goal)

Solution is then defined as a set of formulas X* which satisfies three conditions:

1. X* is a subset of (K union T) [as before]
2. X* is consistent [as before]
3. Every member of G can be deduced from X* [as before]
4. **For every task in X*, there is a role which is responsible for it [new]**

Observation 3:

RE involves both problem-solving and decision-making.

Applying the guidelines: Taxonomy of solution concepts influenced by formalism features

Suppose we now add:

- Preference relations between typed formulas
- Distinction between Mandatory goals (have to be satisfied) and other goals

Solution is then defined as a set of formulas X* which satisfies three conditions:

1. X* is a subset of (K union T) [as before]
2. X* is consistent [as before]
3. **Every mandatory goal can be deduced from X* [new]**
4. For every task in X*, there is a role which is responsible for it [as before]
5. **No set Y satisfies conditions 1, 2 and 3, and ranks higher than X* [new]**

Observation 4:

The more expressive a formalism, the more decision-making it can require during modeling.

Observation 4:

The more expressive a formalism, the more decision-making it can require during modeling.

- Consider this requirement:

  "Access to video-on-demand should be granted to all paying users."

- How many decisions does one need to make to record it in:
  - Propositional logic?
  - Propositional logic with types over formulas?
  - First-order logic?
  - First-order logic with time?

Observation 4:

The more expressive a formalism, the more decision-making it can require during modeling.

Guideline H

Design the formalism so that it is made of modules which group features relevant to a particular step of RE, and/or a particular audience for requirements models.

Observation 4:

The more expressive a formalism, the more decision-making it can require during modeling.

Applying the guidelines:
Ongoing work on the Techne family of formalisms

Table 1: Overview of the Techne formalisms according to their components; Symbol "•" reads "present", "○" reads "absent".

Components		T1 (§5)	T2 (§6)	T3 (§7)	T4 (§8)	T5 (§9)
Core modalities	Goal	•	•	•	•	•
	Domain assumption	•	•	•	•	•
	Task	•	•	•	•	•
	Quality constraint	○	○	○	○	•
	Softgoal	○	○	•	•	•
	Soft dom. assumpt.	○	○	•	•	•
Optionality modalities	Mandatory	○	○	○	•	•
	Preferred	○	○	○	•	•
	Inherited	○	○	○	•	•
Agency modalities	Agent	○	•	•	•	•
	Role	○	•	•	•	•
Primitive relations	Inference	•	•	•	•	•
	Conflict	•	•	•	•	•
	Responsibility	○	•	•	•	•
	Ability	○	•	•	•	•
	Occupancy	○	•	•	•	•
	Commitment	○	•	•	•	•
	Preference	○	○	•	•	•
Variables	Binary Boolean	•	•	•	•	•
	Rational	○	○	○	○	•

Summary of the observations

Observations:

1. Different stakeholders have different expertise.
2. Different stakeholders have different decision authority.
3. RE involves both problem-solving and decision-making.
4. The more expressive a formalism, the more decisions it can require during

Summary of the guidelines

Guidelines:

A. Propositional formalisms preserve statements of requirements in natural language. This makes them more convenient for non-engineering target audiences than first-order formalisms.

B. Types of formulas in a formalism should reflect categories relevant to the target audience.

C. How inconsistency should be tolerated depends on what inconsistency should convey in problem-solving and decision-making.

D. How inconsistency is tolerated needs to be decided when the formalism is being designed.

E. A formalism for RE needs to incorporate a taxonomy of problem and solution concepts.

F. A formalism for RE needs to include features allowing the comparison of solution fragments and this over various criteria.

G. A formalism for RE needs features used to aggregate comparisons of fragments of solutions, i.e., features that allow the definition of decision rules.

H. Design the formalism so that it is made of modules which group features relevant to a particular step of RE, and/or a particular audience for requirements models.