Ontologies and rule-based knowledge in Knowledge-Driven Optimization

Piotr Macioł

Andrzej Macioł

AGH University of Science and Technology, Krakow

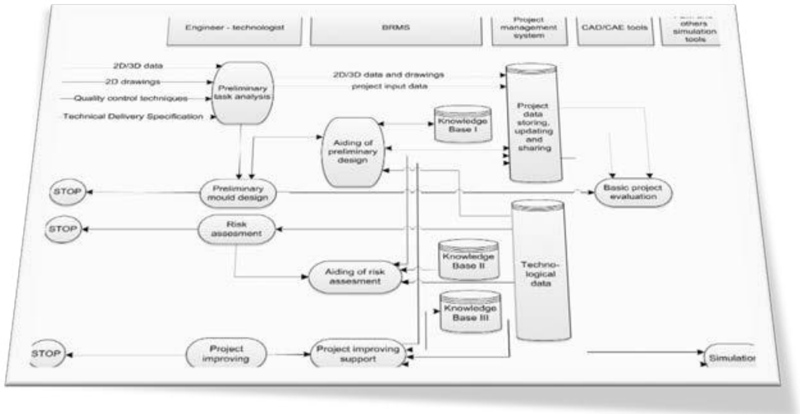
pmaciol@agh.edu.pl

Agenda

- When and why semantic description matters?
- Interoperability of numerical models
- How to combine numerical models and Decision Support Systems
- Examples
- Conclusions

Why semantic matters?

- » Solving a single, isolated “material issue” is *relatively* simple.
- » Solving real engineering problems requires concurrent solving of several coupled models, covering material and business issues.
- » In all realistic cases, different tools must be integrated:
  - – numerical models
  - – optimization frameworks
  - – business decision support systems (BDSS)
  - – manufacturing systems
  - – data analysis systems
  - – and others …



AGH

Why semantic matters?

- Tools differs with data formats, variables meanings, assumptions, boundary conditions etc.
- A single mistake may waste all efforts (Mars Climate Orbiter, 1999)
- Compatibility on a “format” (data file, database, API) level requires some human activities
  - expert knowledge
  - repeating the same work many times
  - error prone
- Computers are not able to “understand” a real meaning of interfacial values
  - e.g effective stress and stress tensor could be treated as completely different variables or as the same variables, both approaches are invalid
- A single, fixed interface would be very stiff and difficult to enforce

Why semantic matters?

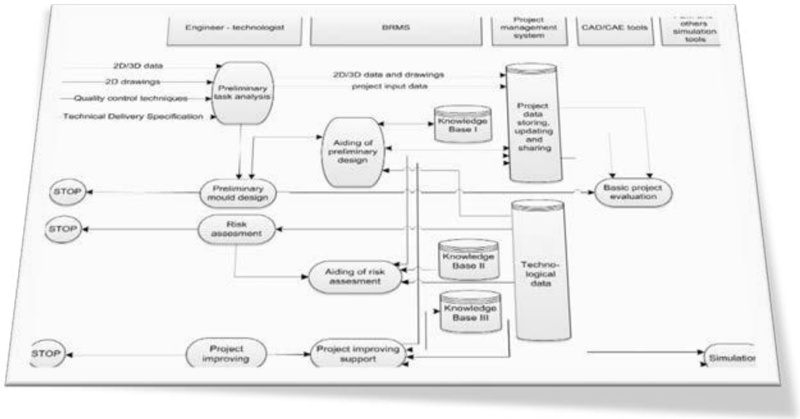
» Analysis of interoperability between two tools is the most reliable if done/verified by a human researcher

» Such trends as:

  – e-market of computational tools
  – an increase of complexity of models with expected reduced designing time
  – flexibility and ability of autonomous modifications of model structures

requires autonomous (with limited or with no human interactions) matching and communication

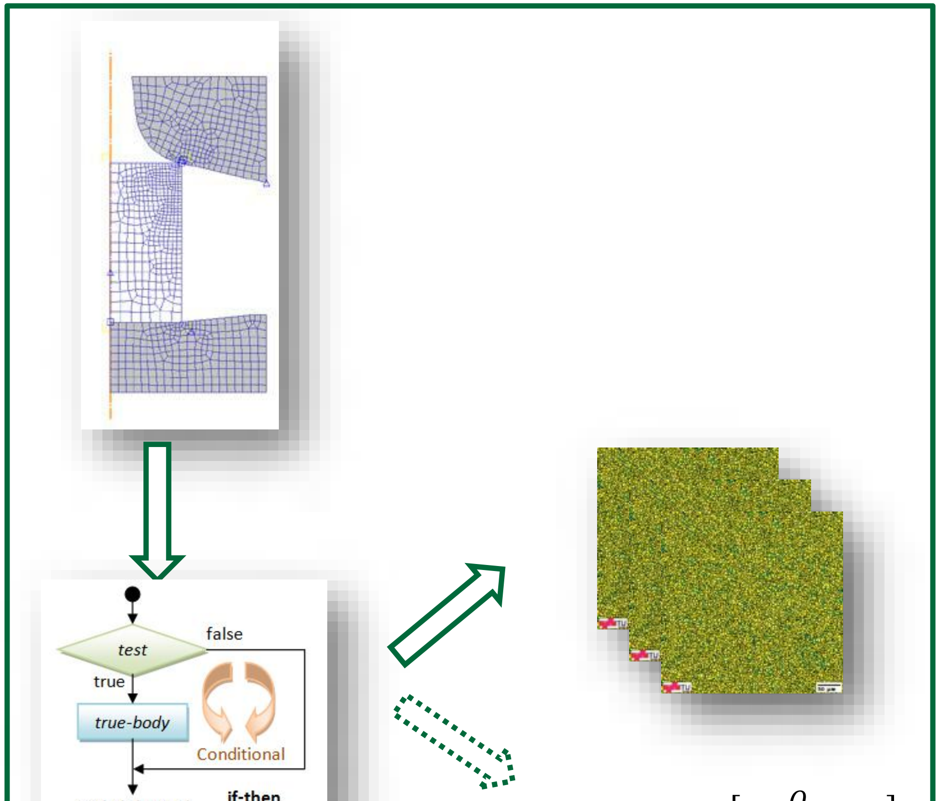
» A single description, understandable by both humans and computers is expected



Interoperability of numerical models

» Case 1: Multiscale/complex numerical models

Flexible, easy-to-design numerical models, including several coupled sub-models, with alternative configurations (also on-line adapting



$$D_{RX} = b_6 \varepsilon^{b_7} \dot{\varepsilon}^{b_8} D_0^{b_9} \exp\left[\frac{Q_{DX}}{R(T+273)}\right]$$

Interoperability
Multiscale/complex numerical models

- A “static” linking of submodels/models/systems can be done by researchers:
  - potentially error-prone
  - require highly skilled, multidisciplinary experts
  - the most flexible
- A “dynamic” linking must be done by artificial agents (computer systems) themselves
  - very fast
  - once developed, multiply used
  - if not supported by “understanding” of numerical/mathematical/material issues, extremally stiff or very unreliable

Forms of representation

» Semantics can be represented with:

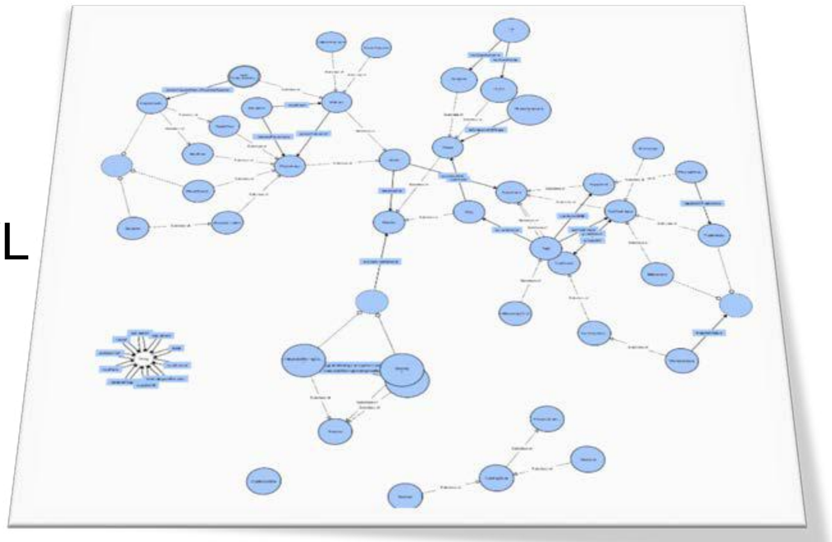
- Propositional Calculus
- Relational Database (closed world assumption)
- First Order Logic (open world assumption, highly expressive, difficult to use)
- Descriptive Logic/ontologies (open world assumption, less expressive, easy to use)

	Statement:	"Mary" "is a citizen of" "France"
	Question:	Is Paul a citizen of France?
"Closed world"	Answer:	No.
"Open world"	Answer:	Unknown.

Example source: Wikipedia

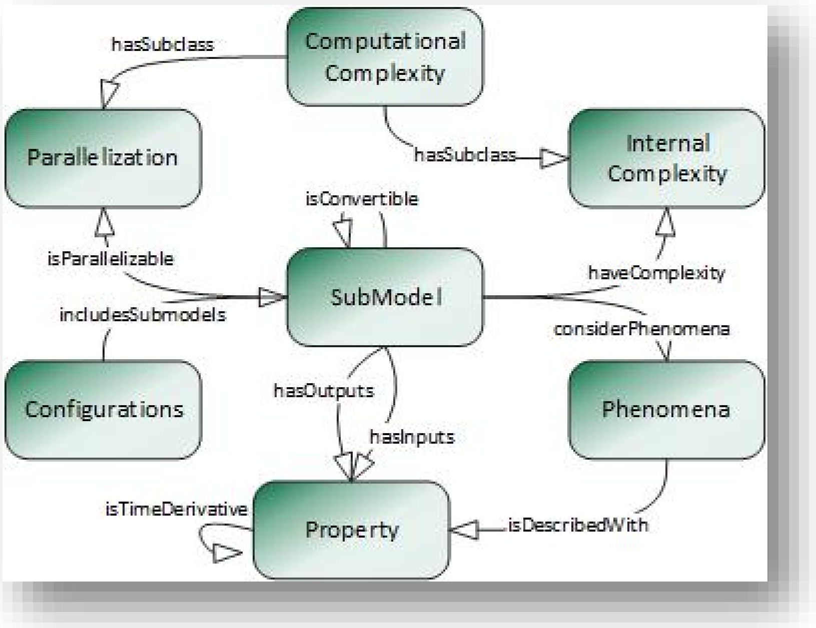
Ontologies

- Not necessary for human researchers (might be useful, however requires additional skills)
- Most common form of a semantic description understandable for computers
- Widely use for communication between computers in B2B solutions (multi-agent systems)
- Provides a structured language, but do not define processes (workflows, guides, etc.)
- A common problem – numerous, dedicated and incompatible ontologies for particular problems
- Ontologies are usually developed using the OWL language which is a higher level language, based on XML and Resource Description Framework (RDF)

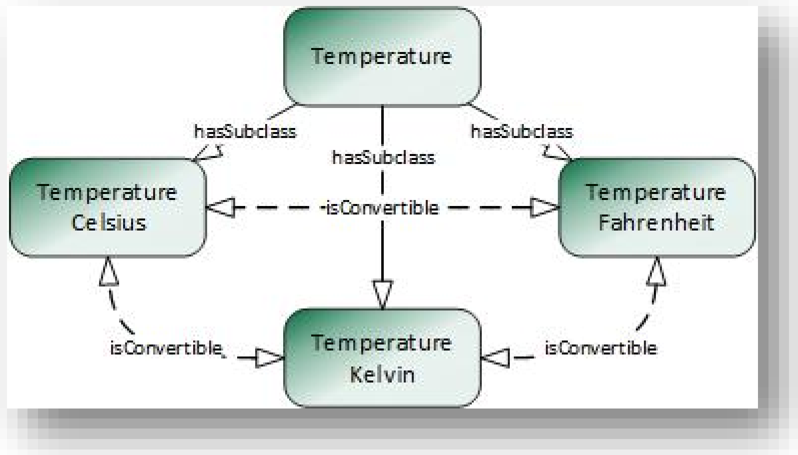


AGH

Multiscale Modeling Formal Ontology (MMFO)

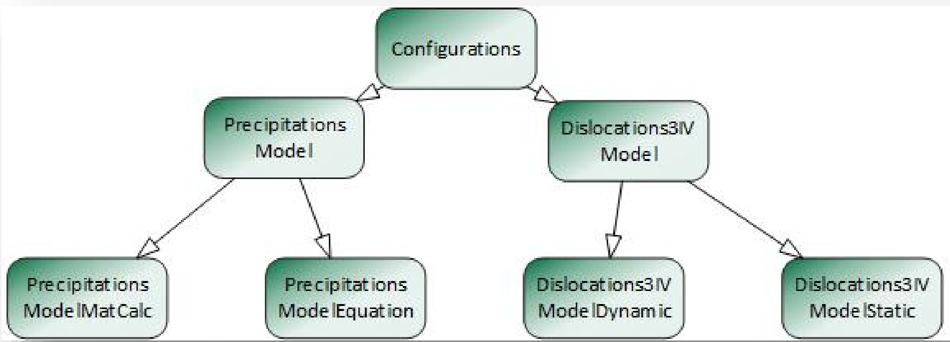


Metaontology – describes what relationships are valid

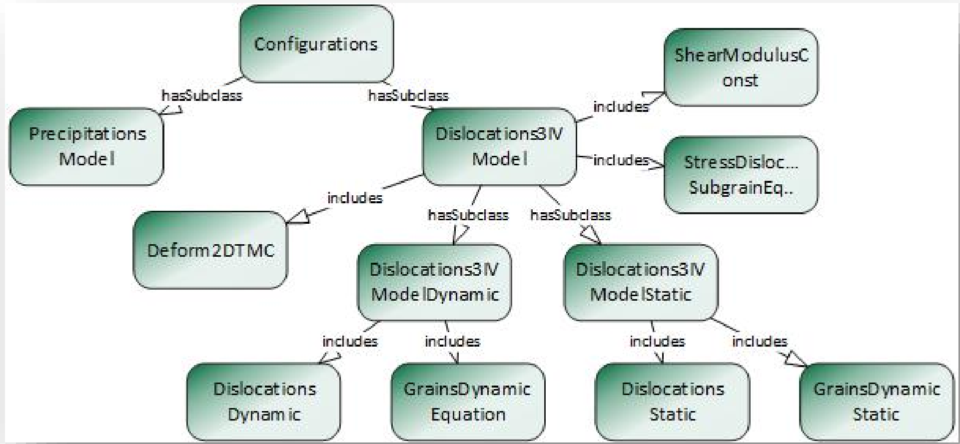


Dependencies between *Properties*

MMFO

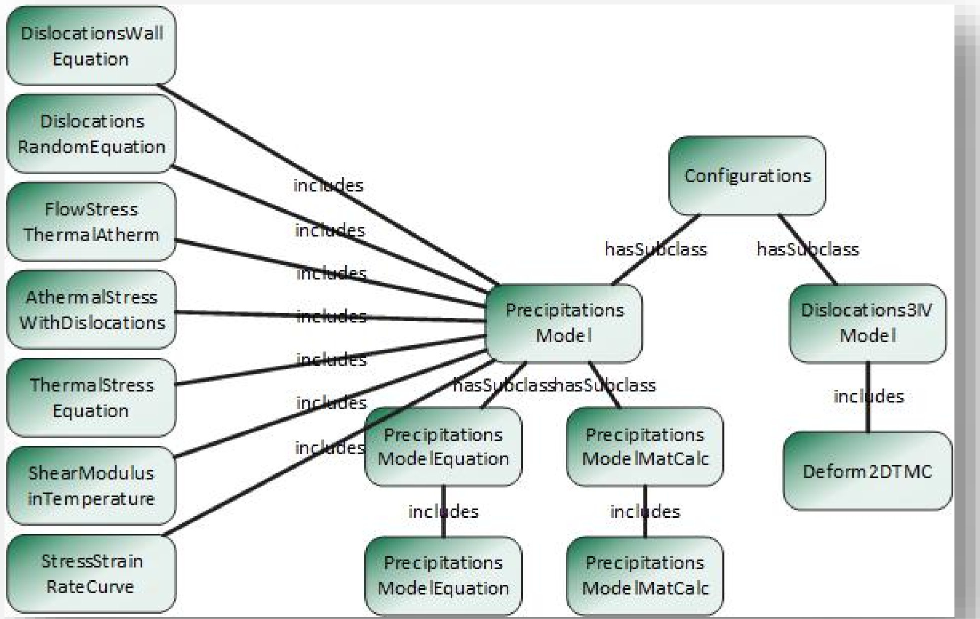


An exemplary hierarchy of Configuration subclasses



Configuration of exemplary model: the three-internal-variable

MMFO



Configurations of exemplary model: precipitation kinetic model

Description logic - OWL based solution

- Advantages
  - Human-readable
  - ***Metaontology*** gives a capability of validation of a syntactic correctness of the submodels description
  - Defines dependencies between submodels and properties/variables (including more complex ones, like *Velocity isTimeDerivative* of *Coordinates*)
  - OWL language is easily convertible to relational databases structures and common formats of data representation (e.g. JSON)
  - OWL based ontology (MMFO) can be easily used a common standard
- Disadvantages
  - The are no direct tools to validate the semantic correctness of submodels structure (but since OWL classes are well defined, it is ease to develop such tools)
  - Developing of ontology is tedious

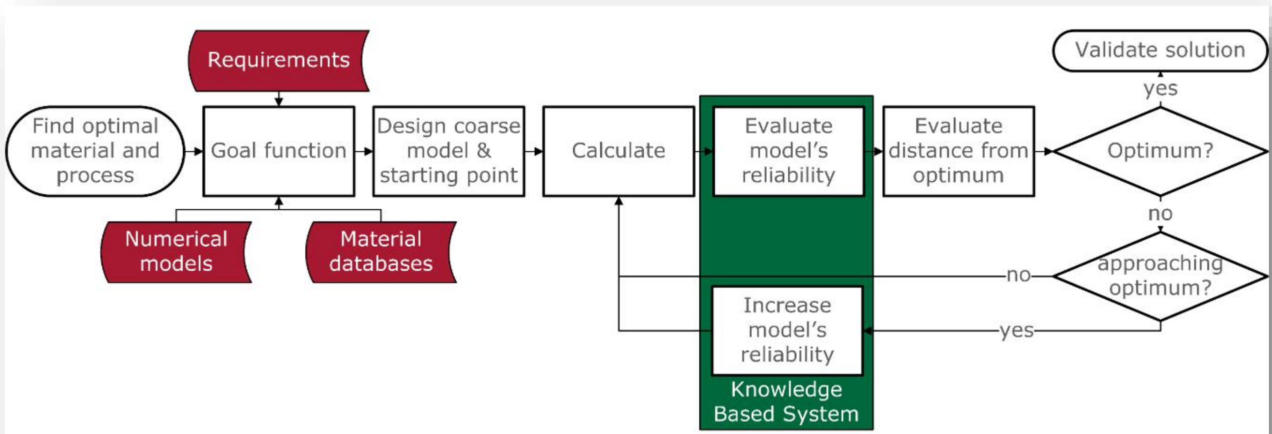
Interoperability between BDSS and numerical tool

Case 2: Knowledge Driven Optimization

» Classical optimization algorithms use only mathematical properties of response surface

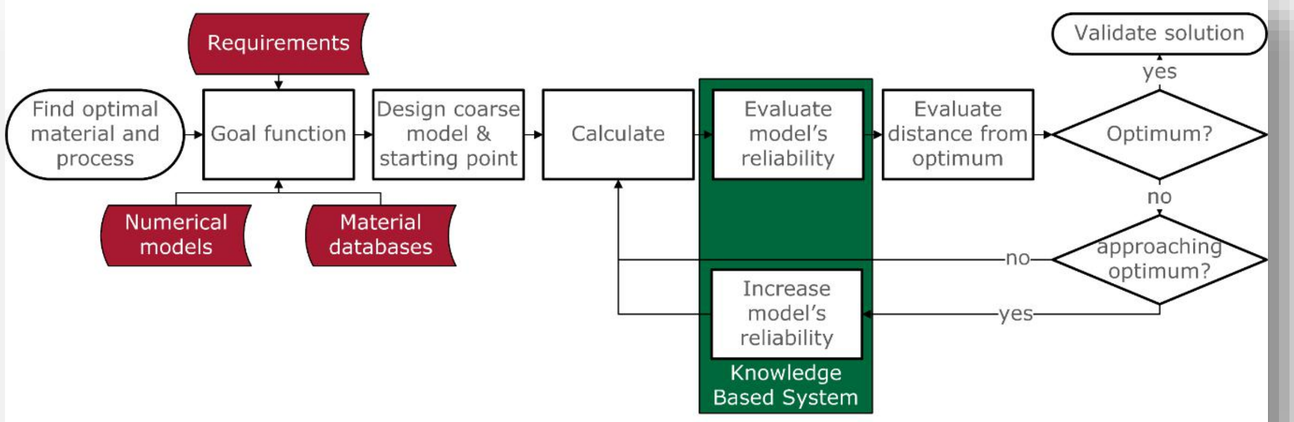
» Heuristic algorithms use additional knowledge (simple formulas, hardcoded or slightly customizable) to speed-up optimization

» Knowledge Driven Optimization uses complex knowledge (rule-based, big-data, separated from algorithm itself)



Combining ontologies and reasoning engines

- An ontology is just a language, do not provide reasoning mechanisms
- Inference engines (based usually on FOL) solves particular, well defined and “closed” problems – they cannot “talk” with external systems
- Using a common ontology simplifies their unambiguous description, but decision-making requires additional integration mechanism
- Such mechanism could be based on IT tools used for modelling and automation of workflows



Interoperability between BDSS and numerical tool examples

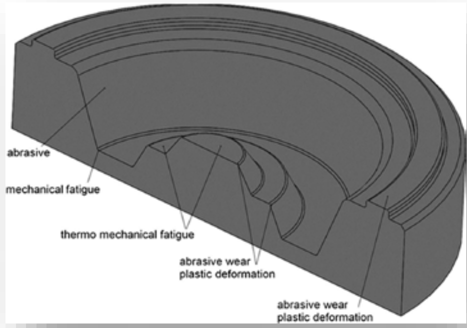
» A choice of most suitable models of forging dies wear

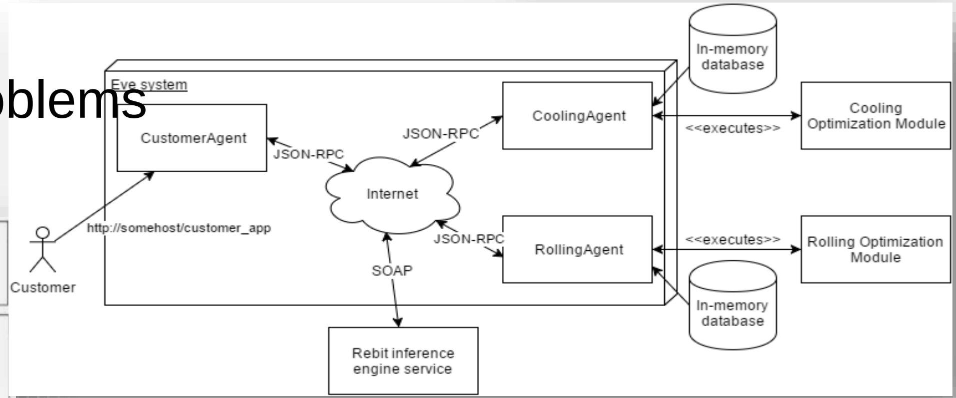
» Acquiring knowledge from numerical models

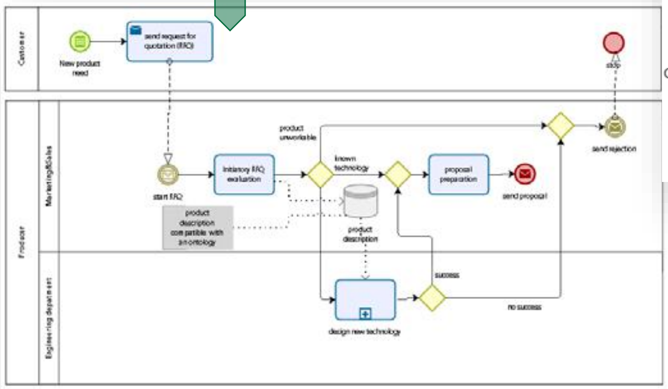
» Multi-agents system for technology optimization

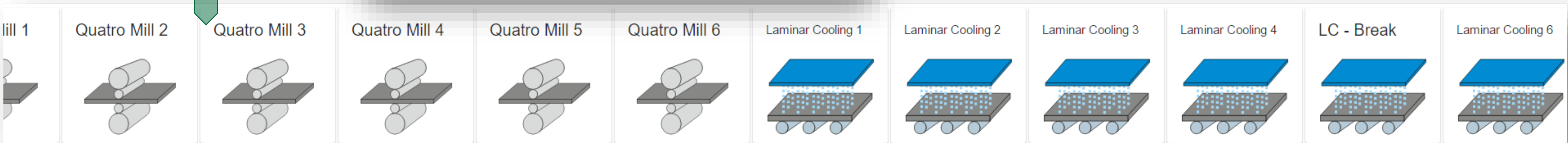
» Request For Quotation evaluation

» Optimization of complex technological problems





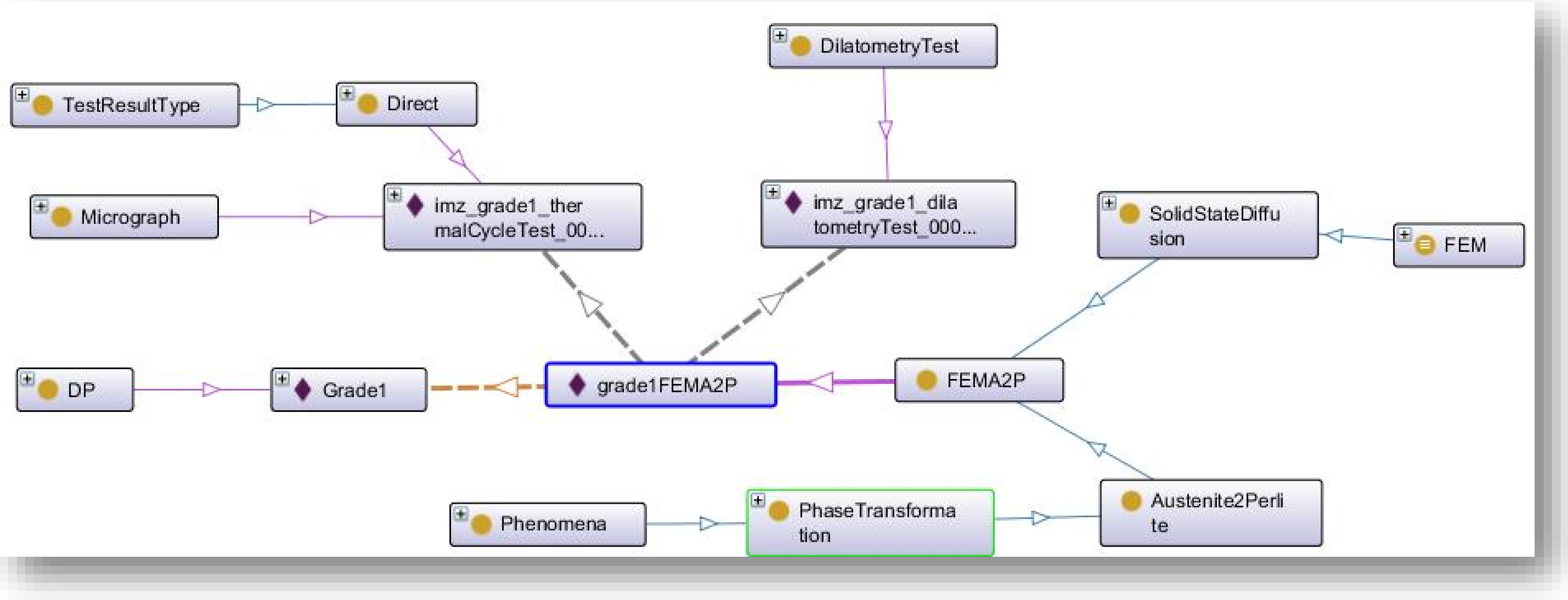




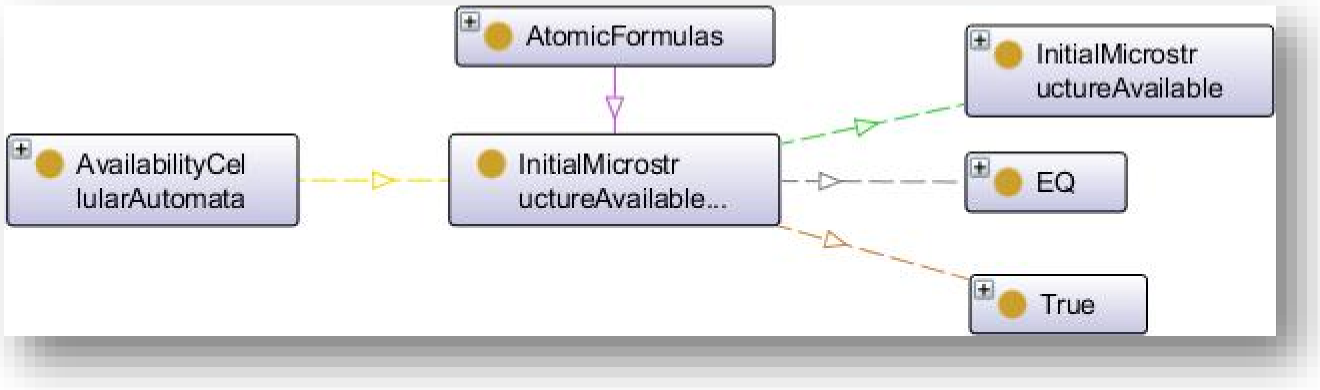
Ontologies and reasoning

- In theory, decision support systems could directly use taxonomies based on ontologies for reasoning
- Reasoning mechanisms developed for Description Logic are less effective than used in First Order Logic (Open/closed world problem)
- Alternatively, an DL-based Knowledge Base can be transformed to FOL representation

A fragment of problem ontology



Translating ontology for atomic formulas



Generated rules

RULE AvailabilityCellularAutomata
IF InitialMicrostructureAvailable = True AND
ComputPowerAvailable = High AND CARulesAvailable = True
THEN CellularAutomataAvailable = True

RULE AvailabilityInitialMicrostructure_01
IF InitialMicrostructureFromMicrograph = True
THEN InitialMicrostructureAvailable = True

RULE MicrographAvailable
IF GradeMicrograph = GradeComputation AND
GrainSizeCompare(MeanGrSi, MeanGrSiComp, Deviation) = True AND
PhaseComposCompare(PhaseCompos, PhaseComposComp) = CloseEnough
THEN InitialMicrostructureFromMicrograph = True

RULE CARules
IF EstimatedProcessTemperature IN CAModelTemperatureRange AND
EstimatedCoolingRate IN CAModelCoolingRateRange AND
MaterialGradeRules = MaterialGradeComputation
THEN CARulesAvailable = True

Conclusions

» Ontologies may be used as an “augmented vocabulary”

» Ontologies simplifies communication between components of software solution

» Using a common ontology still requires translation operations

» Decision making with knowledge defined with ontologies is ineffective

» “Ontology-based” knowledge can be transformed into rules (for expert systems)

» A family of common ontologies should be agreed

» A risk of overcomplication is significant!