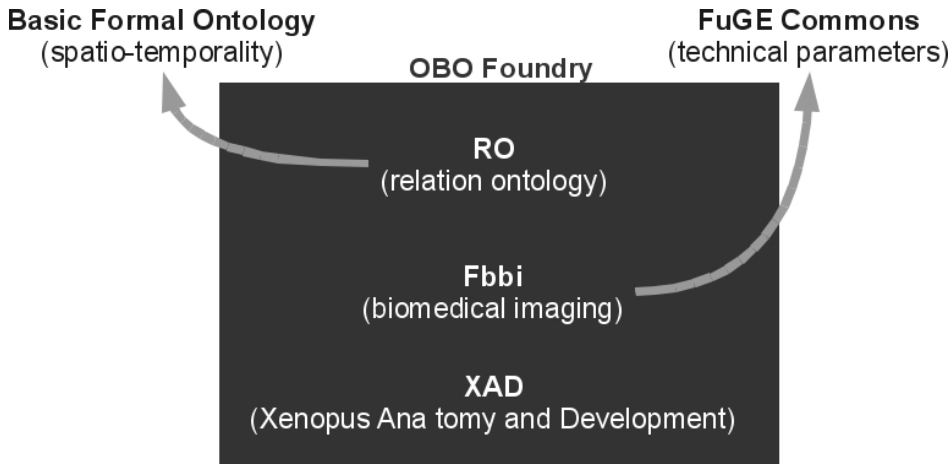
Constraint management in engineering of complex information systems

MN. Terrasse, M. Savonnet, E. Leclercq,
P. Naubourg, A. Da Costa, and T. Grison
marie-noelle.terrasse@u-bourgogne.fr

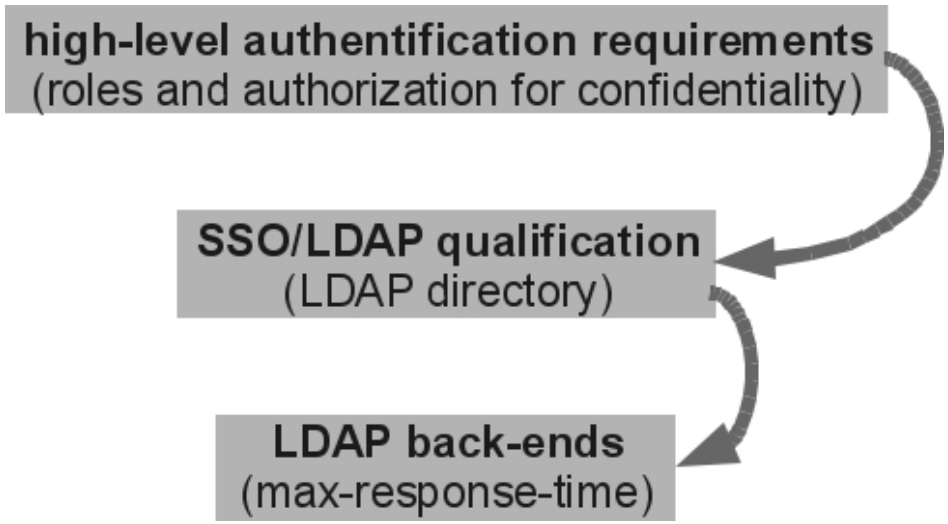
Constraints in information systems with distributed and virtual organizations (1)

- A gap between the semantical and operational organizations:
  - Semantical units of knowledge
    small-grain, highly inter-related
    abstract constraints, i.e., compliance, dependencies



Constraints in information systems with distributed and virtual organizations (2)

- A gap between the semantical and operational organizations:
  - Semantical units of knowledge
    small-grain, highly inter-related
    abstract constraints, e.g., compliance, dependencies
  - Operational units of resources
    coarse-grain, automatically combined
    constraints at various abstraction levels



Constraints in information systems with distributed and virtual organizations (3)

- A gap between the semantical and operational organizations:
  - Semantical units of knowledge
    small-grain, highly inter-related
    abstract constraints
  - Operational units of resources
    coarse-grain, automatically combined
    constraints at various abstraction levels
- Engineering processes:
  - "ETL-like" use of resources:
    - **E** selection of relevant resources
    - **T** definition of organization criteria
    - **L** resource integration
  - MDE-approaches
    models at various abstraction levels
- **Constraints and architectures of models ?**

Towards a typology of constraints

modeling paradigm	constraints on open-world and close-world assumptions	application: PostgreSQL, protégé

modeling language	constraints on multiplicities of multiple associations on one class (UML/OCL)	`context faceted-object`
		`inv composition:`
		`not ( self.constains1->isEmpty()`
		`and self.constains2->isEmpty())`

business context	*functional* CNIL-related constraints in France	`a Social Security number cannot be used as a primary key`

	non-functional
	constraints due to security requirements
		secureUML:
		`<<secureuml.permission>>`
		`<<secureuml.role>>`

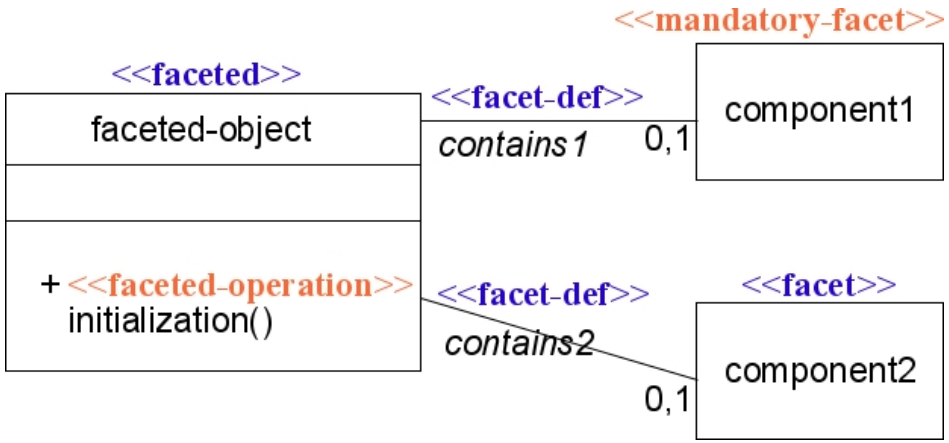
application context	*functional* multibase search for a protein sequence	`mandatory use of the Uniprot assession numbers`
	non-functional an application 's web-site must have wide-scope	`forbidden use of XHTML features`

Abstraction versus finding, formulating, validating constraints

Finding and formulating constraints

- Ontology-based approaches:
  "finding" is an incremental process
  uniform "formulation" carried out at all ontological levels
- Model-based approaches:
  - "finding" with **semi-automatic constraint generators**
    - Costal, 2008
      typology of constraints (e.g., path-based, value-based)
      typology of operations (e.g., on instances, on relations)
    - model-checking
      SCR (Heitmeyer & al.), GME (Nordstrom & al.)
  - uniform "formulation" of **structural constraints** (OCL)
  - "formulation" of **behavioral constraints**
    - OCL *pre-* and *post-* conditions
    - OCL *body* clauses
    - Action Semantics Languages and code snippets
      e.g., EP Kelsen, 2007, ASOQ Hausten, 2004

Polymorphous constraints
a 3D-space (construct, diagram, abstraction level)



- Pre-condition (class diagram)

  context faceted-object::initialization()

  pre: self.contains1->notEmpty()
- guarding condition (statechart)

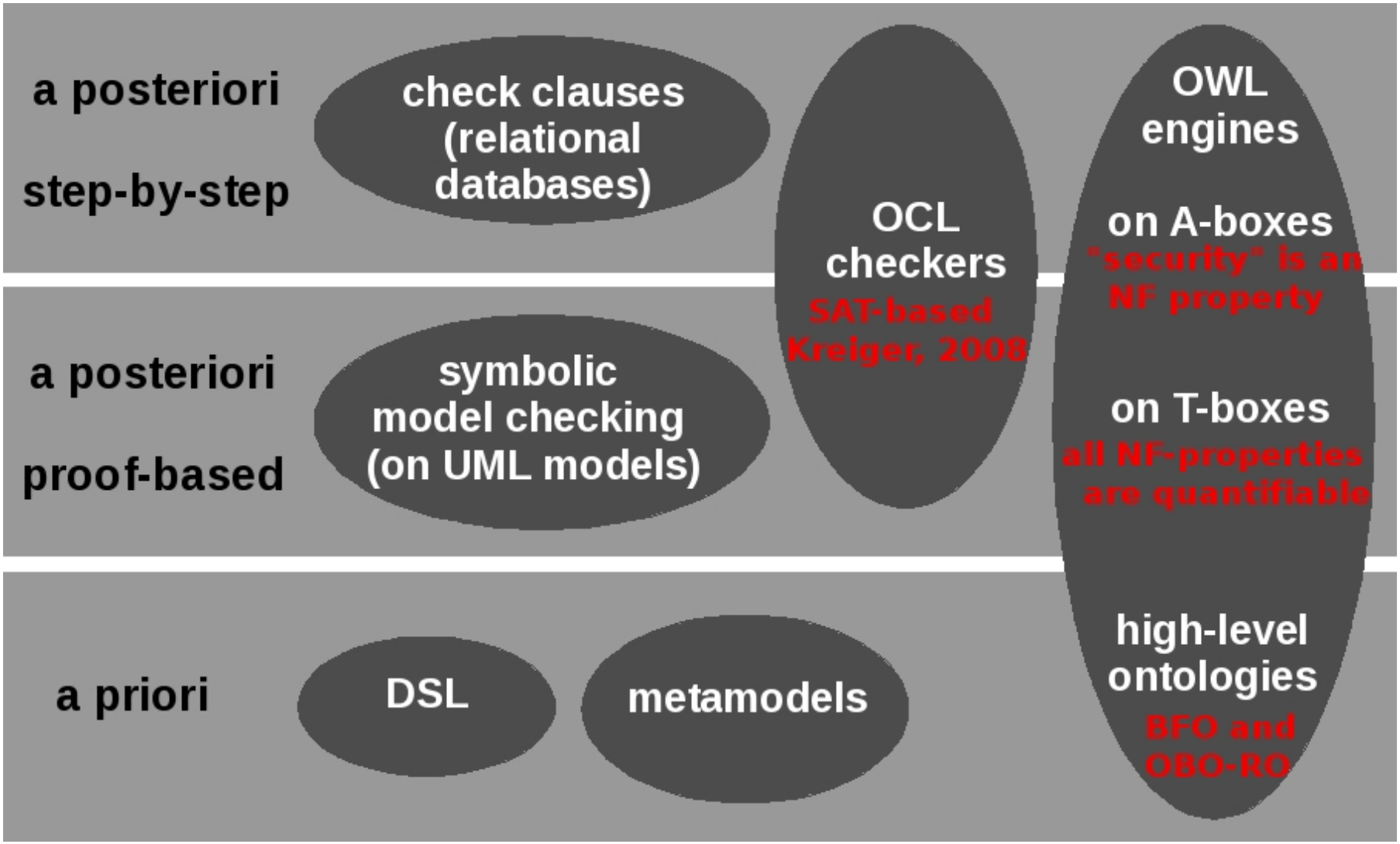
  self.contains1->notEmpty()/initialization()
- Stereotypes

  class : facet, faceted, **mandatory-facet**

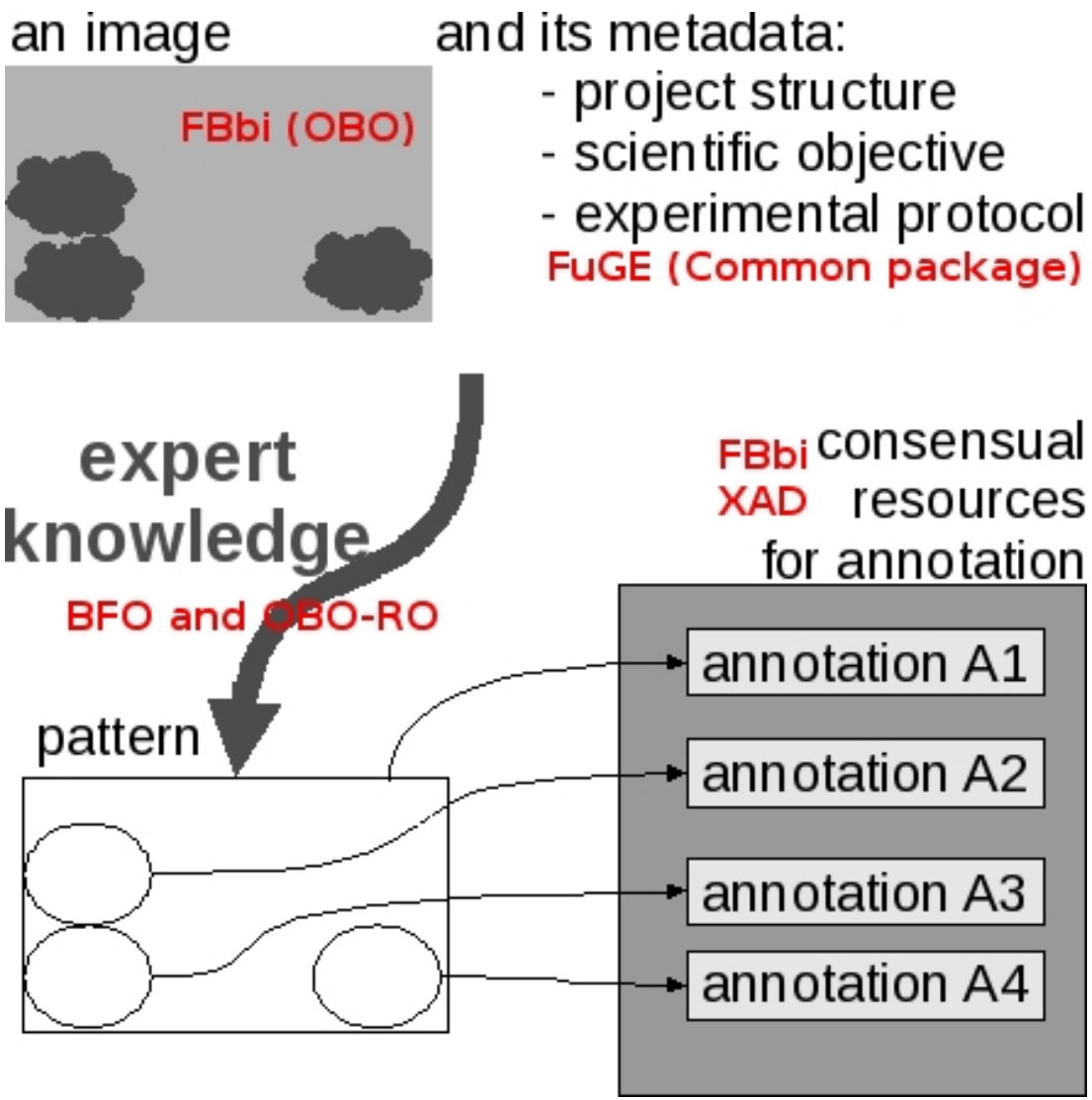
  association : facet-def

  operation : **faceted-operation**

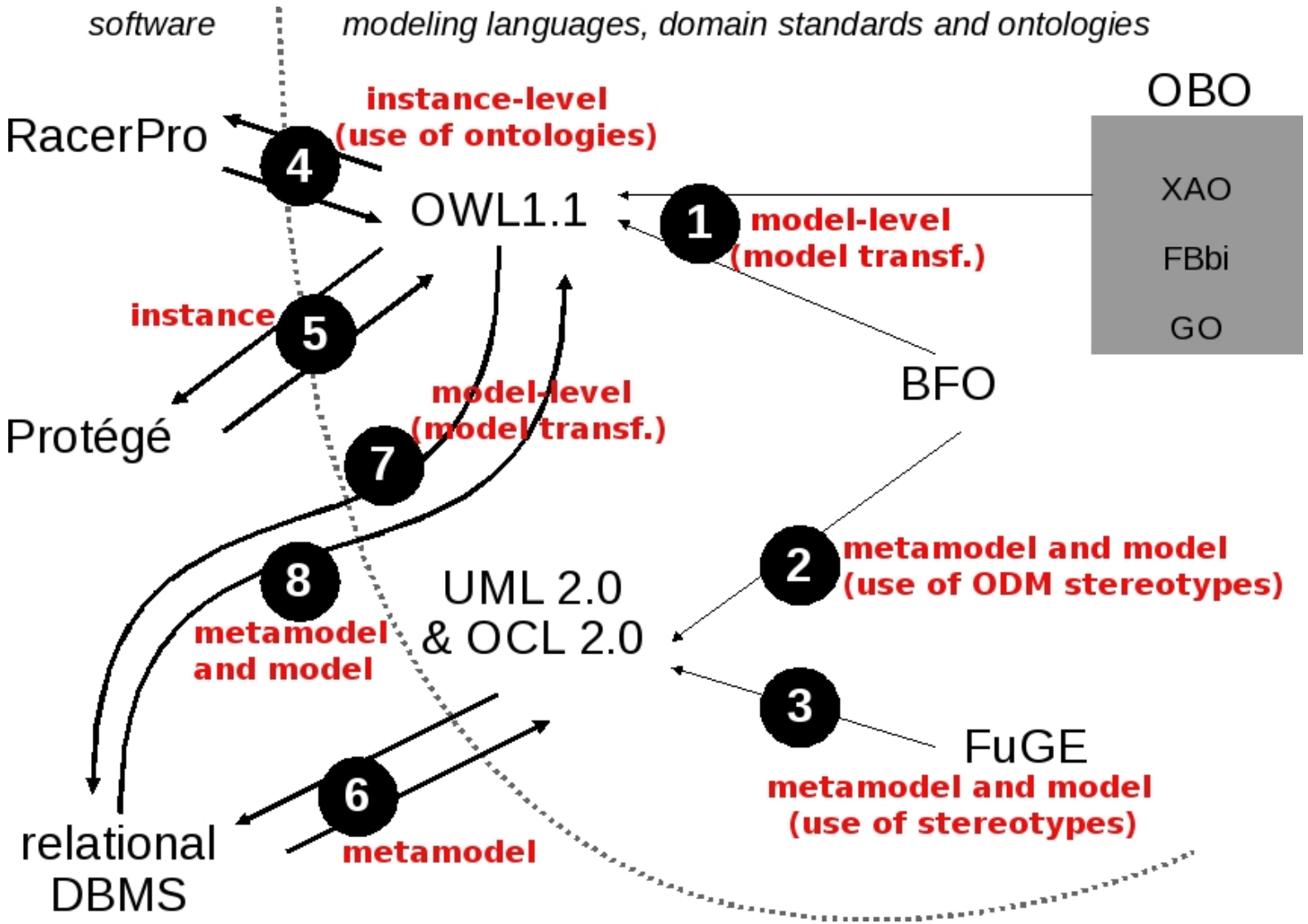
Constraint verification



An illustrative exemple: XenOnt



The XenOnt's constraints



Conclusion

- A 3D-space for constraint management
  in Model Driven Engineering
  combination of formal methods Clarke, 96 Lamsweerde,00
  availability of relevant frameworks
- A major issue
  constraint **traceability**
  - while changing abstraction levels
  - during model transformations
- A DRE-perspective
  early constraint verification
  high-level ontologies, metamodels and DSLs
  enforcing the use of constructs that embed constraints
  might be easier in domain-specific environments