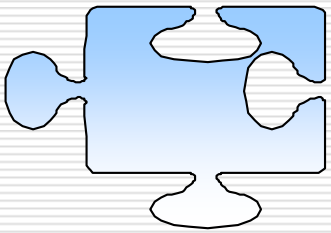
Dealing with Variability within a Family of Domain Specific Languages

Comparative Analysis of Different Techniques

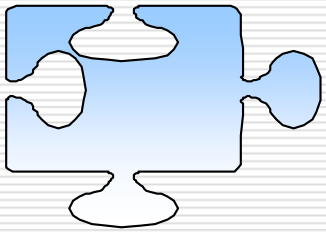
Ileana Ober, Louis Féraud, Christian Percebois

Source problem

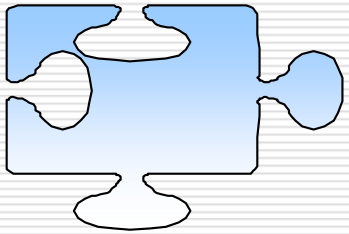
Related (yet different) languages need to be used together



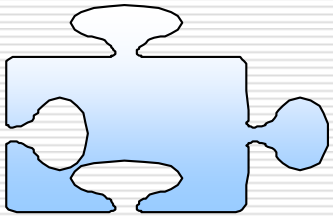
Pluto



Mois



Elisa

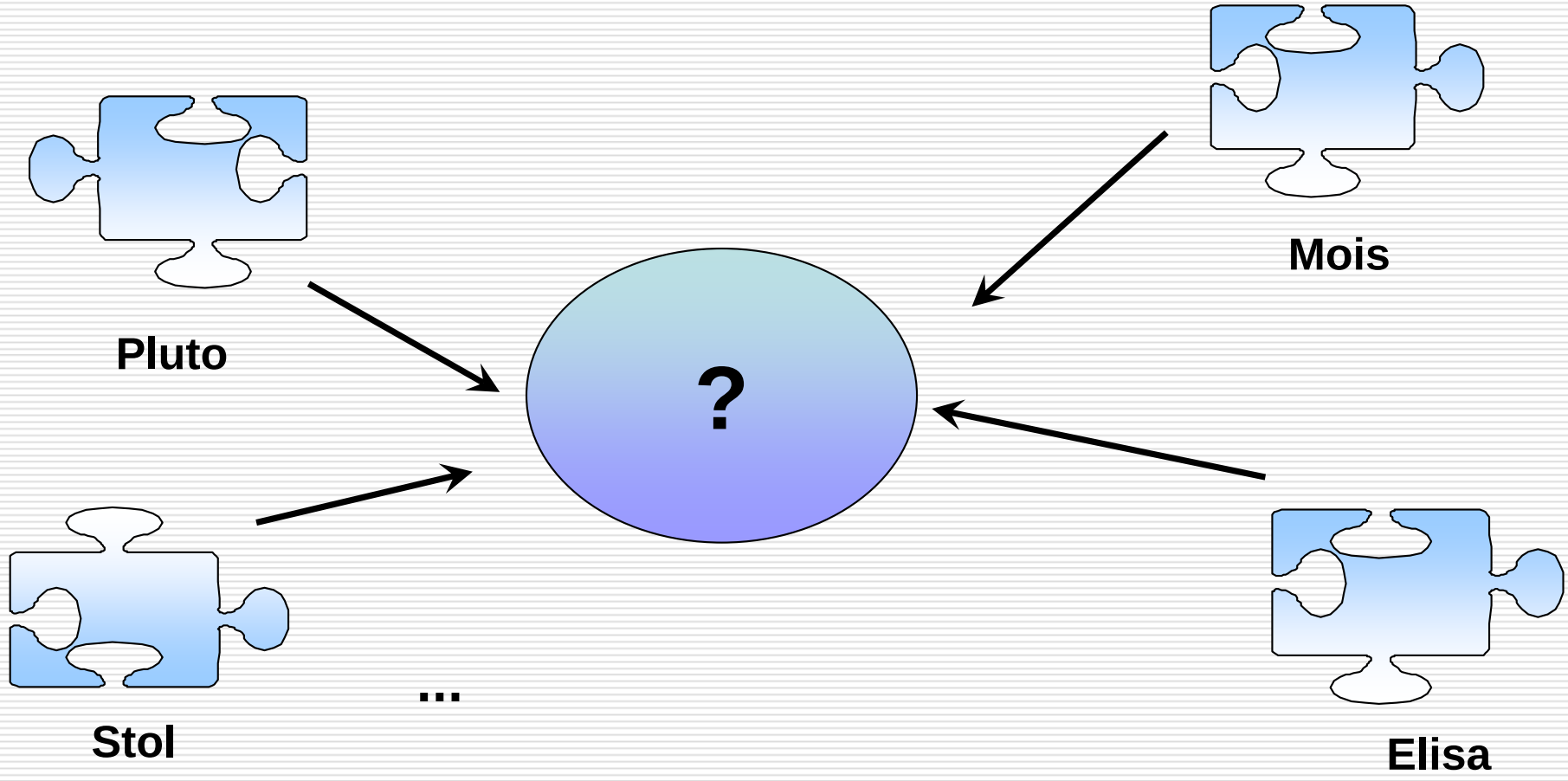


Stol

...

Specific spacecraft operations languages

What approach for interoperability ?



Overview

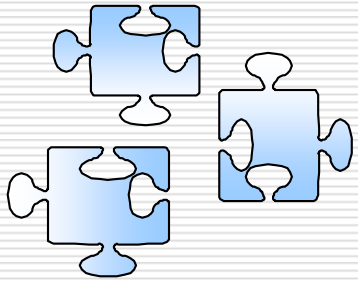
- Family of DSLs
- Unifying a family of DSLs
- Discussion
- Conclusion

Domain Specific Language

- Abstracts the concepts of a business domain
- Is a specialized and problem-oriented language
- Is accessible to domain engineers
- Has a reduced size
- Often combined with transformation tools

Family of DSLs

- Languages that cover a same domain
- Several DSLs for a specific domain
  - Similar concepts and operations
  - Similar hypotheses and requirements
  - Syntactical and semantical variations

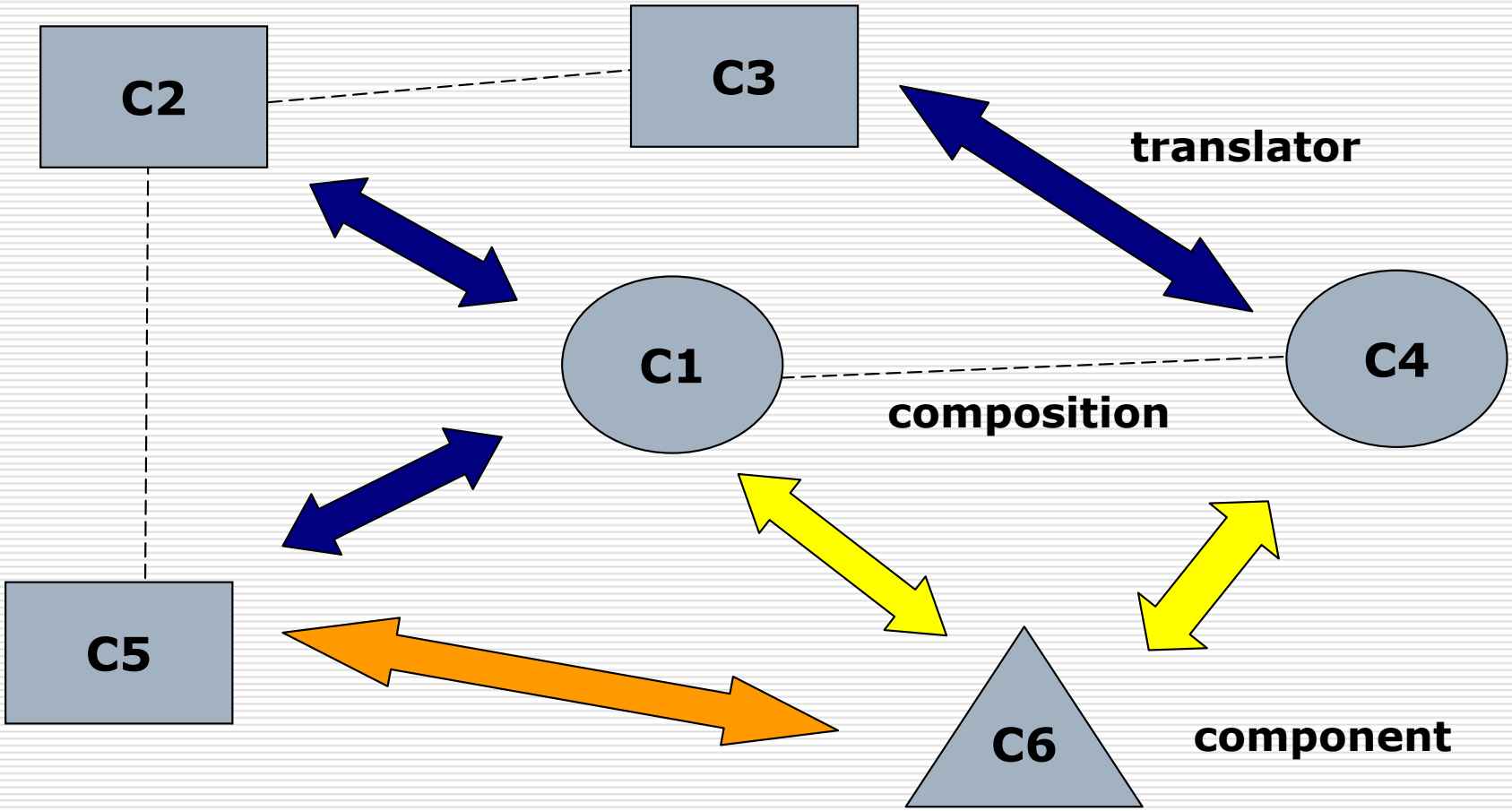


- Related concepts
- Focus on different aspects

Unifying a family of DSLs

- Various solutions
  - Traditional approach
  - Model based approach
  - Problem specific approach
  - Mixed approach

Traditional approach



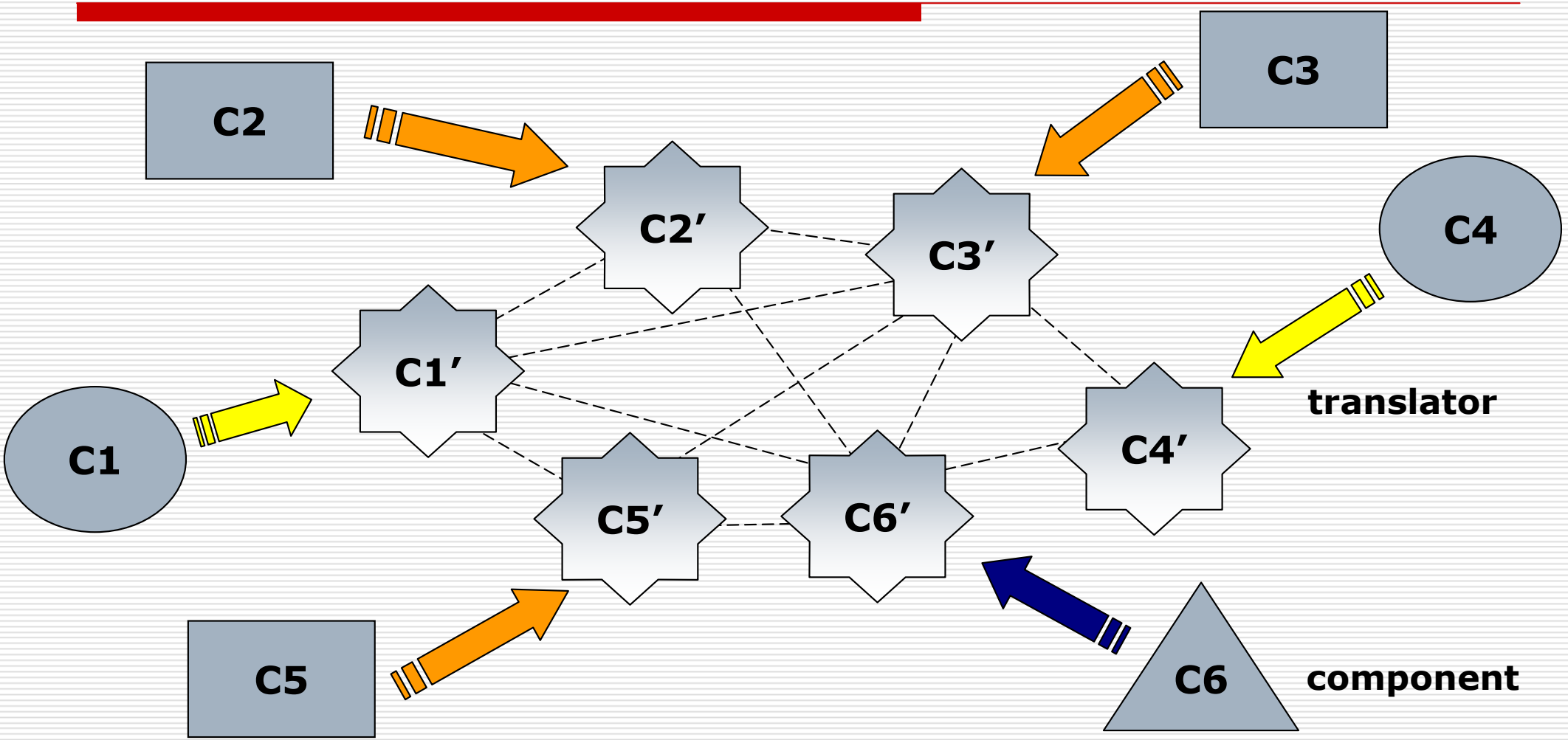
Traditional approach

- Advantages
  - Grammar engineering reuse

    attribute grammars, environments for languages definition (ASF+SDF)

- Drawbacks
  - Specific translator for each language pair
  - $O(n^2)$ complexity

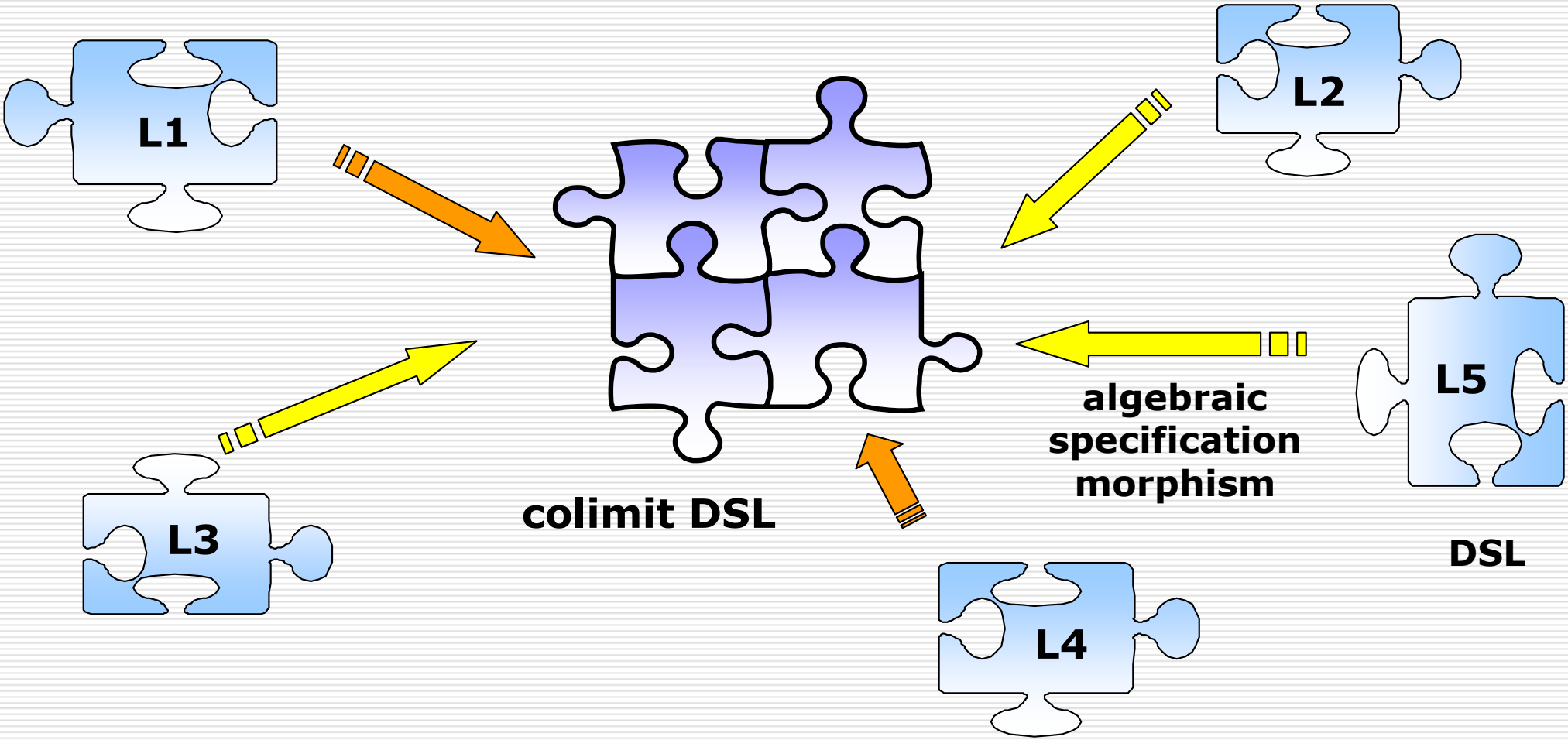
Model based approach



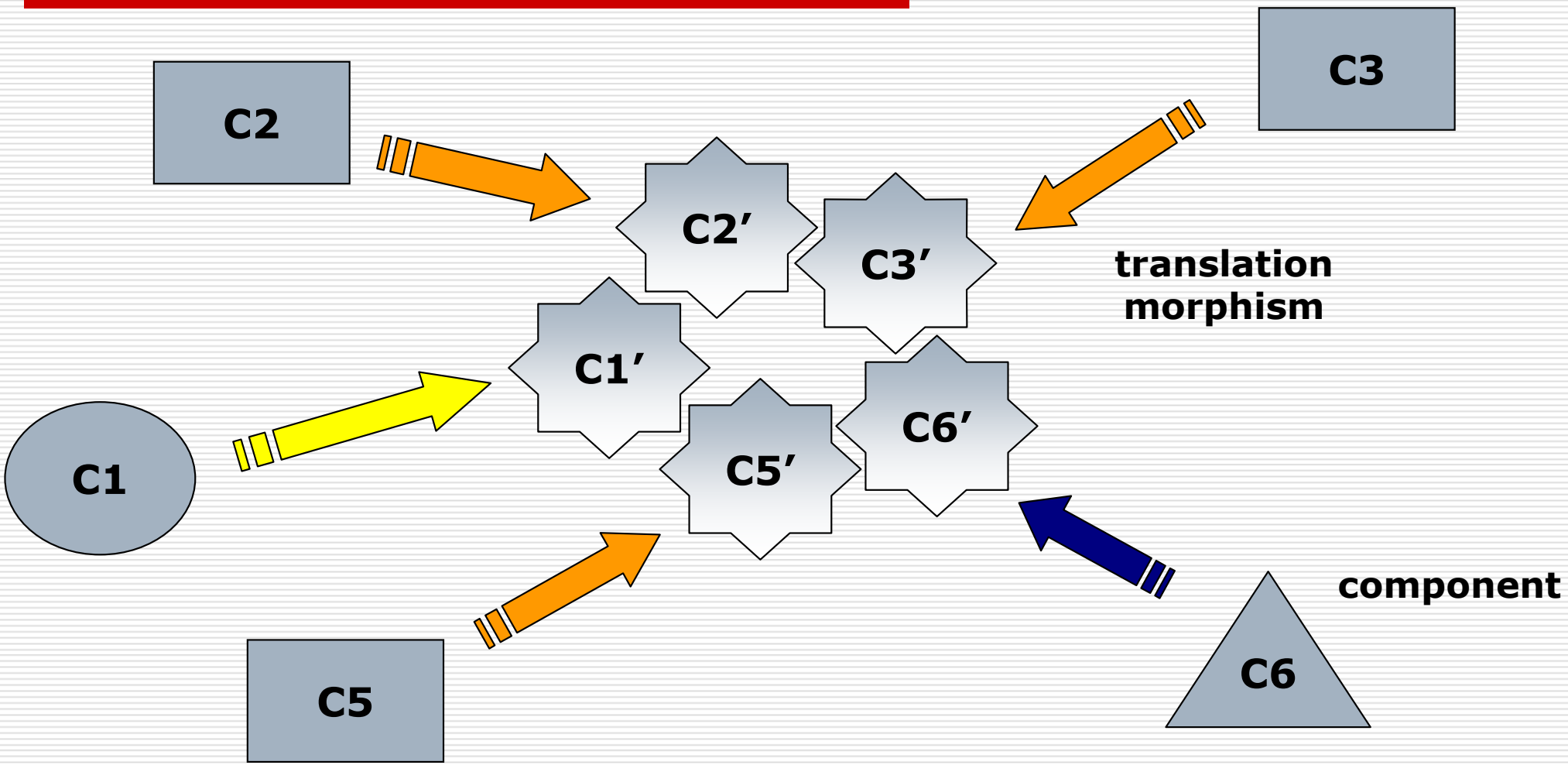
Model based approach

- Advantages
  - Explicit relationships between concepts
  - O(n) complexity
- Drawbacks
  - Difficult consensual definition of an unifying model

Problem specific approach



Problem specific approach

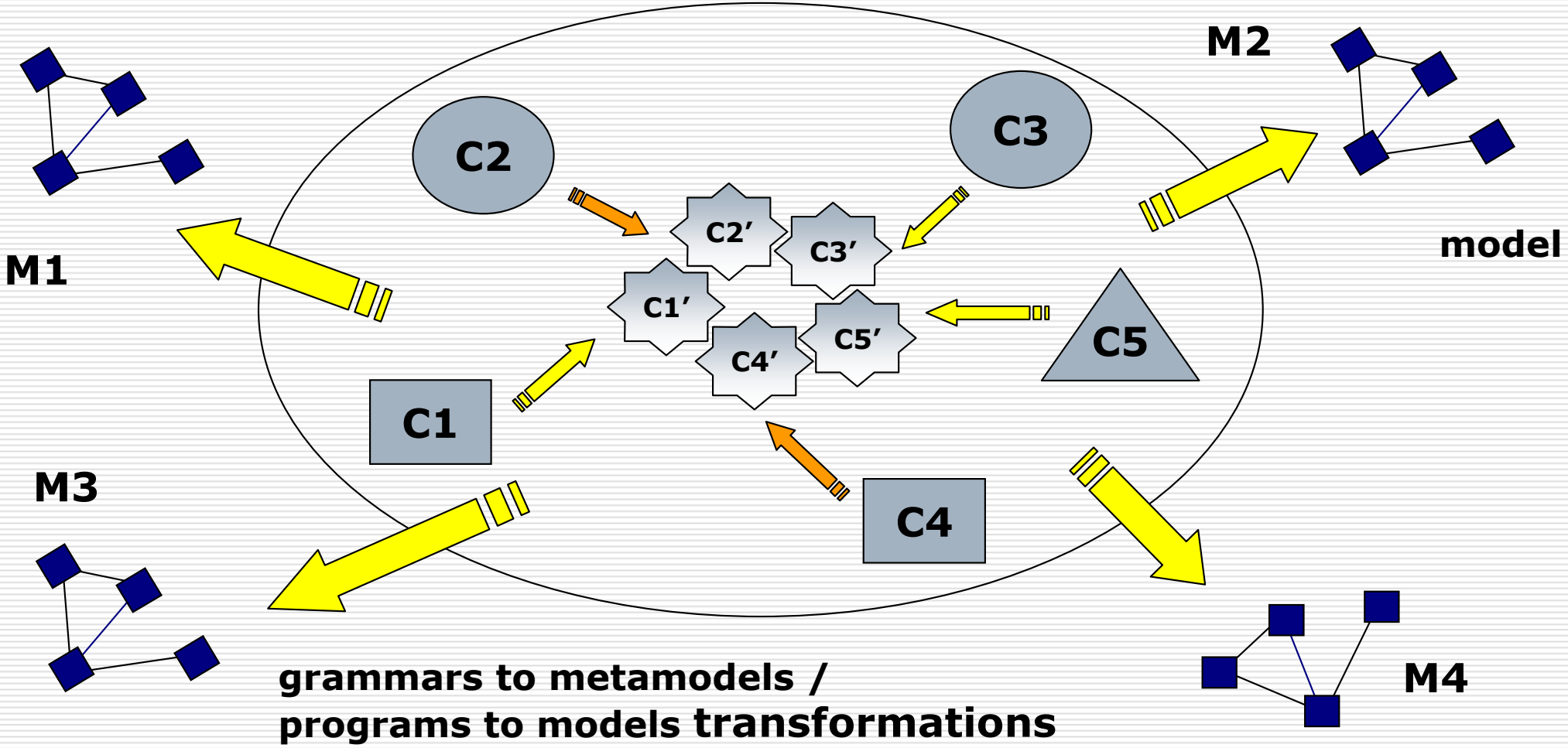


Problem specific approach

- Advantages
  - Formal definition of an unifying language

    algebraic specifications, category theory, colimits…
  - Properties preservation
- Drawbacks
  - Defining a matching between related concepts

Mixed approach



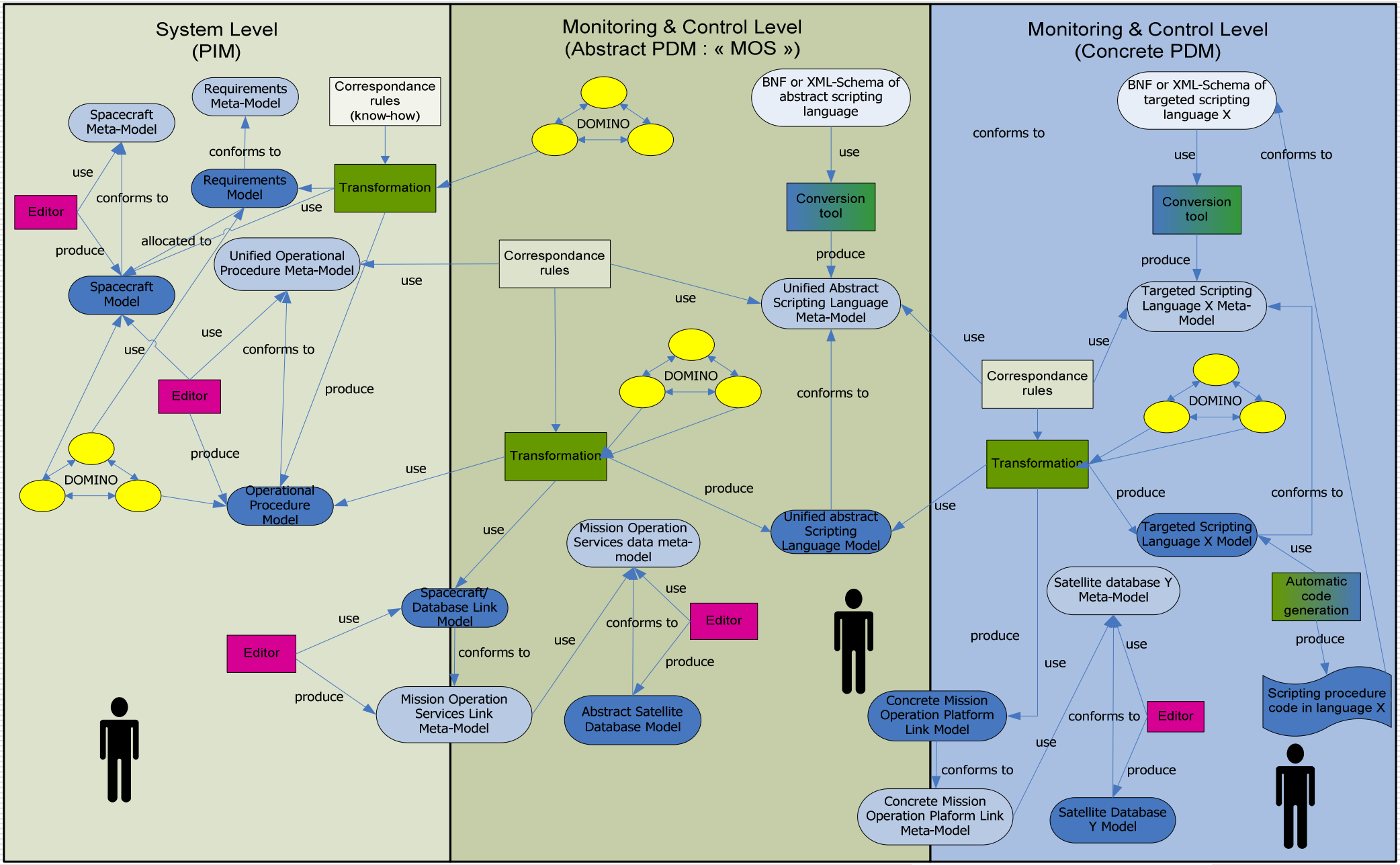
Mixed approach

- Advantages
  - Global view of the domain
  - Combines modeling techniques with a domain specific approach
- Drawbacks
  - Difficult definition of the right domain level information

Discussion… some influencing factors

- Project stage
  - Important reuse parts
- Development team composition
  - Capitalisation of domain engineers expertise
- Degree of variability
  - Draft RFP Common Variability Language (CVL) Technical Report, OMG, 2009

Conclusion


System Level
(PIM)
Monitoring & Control Level
(Abstract PDM : « MOS »)
Monitoring & Control Level
(Concrete PDM)
Requirements Meta-Model
Correspondance rules (know-how)
Spacecraft Meta-Model
conforms to
use
Requirements Model
Transformation
Editor
conforms to
use
allocated to
produce
Unified Operational Procedure Meta-Model
Spacecraft Model
use
use
conforms to
Editor
produce
produce
DOMINO
Operational Procedure Model
use
DOMINO
BNF or XML-Schema of abstract scripting language
use
Conversion tool
produce
Correspondance rules
use
Unified Abstract Scripting Language Meta-Model
DOMINO
conforms to
Transformation
produce
Unified abstract Scripting Language Model
use
Mission Operation Services data meta-model
use
Spacecraft/ Database Link Model
use
conforms to
Editor
Editor
use
conforms to
produce
produce
Mission Operation Services Link Meta-Model
Abstract Satellite Database Model
BNF or XML-Schema of targeted scripting language X
conforms to
use
conforms to
Conversion tool
produce
Targeted Scripting Language X Meta-Model
use
use
Correspondance rules
DOMINO
Transformation
produce
conforms to
Targeted Scripting Language X Model
use
Satellite database Y Meta-Model
Automatic code generation
produce
produce
use
use
Concrete Mission Operation Platform Link Model
conforms to
Editor
use
conforms to
produce
Concrete Mission Operation Plaform Link Meta-Model
Satellite Database Y Model
Scripting procedure code in language X

Questions