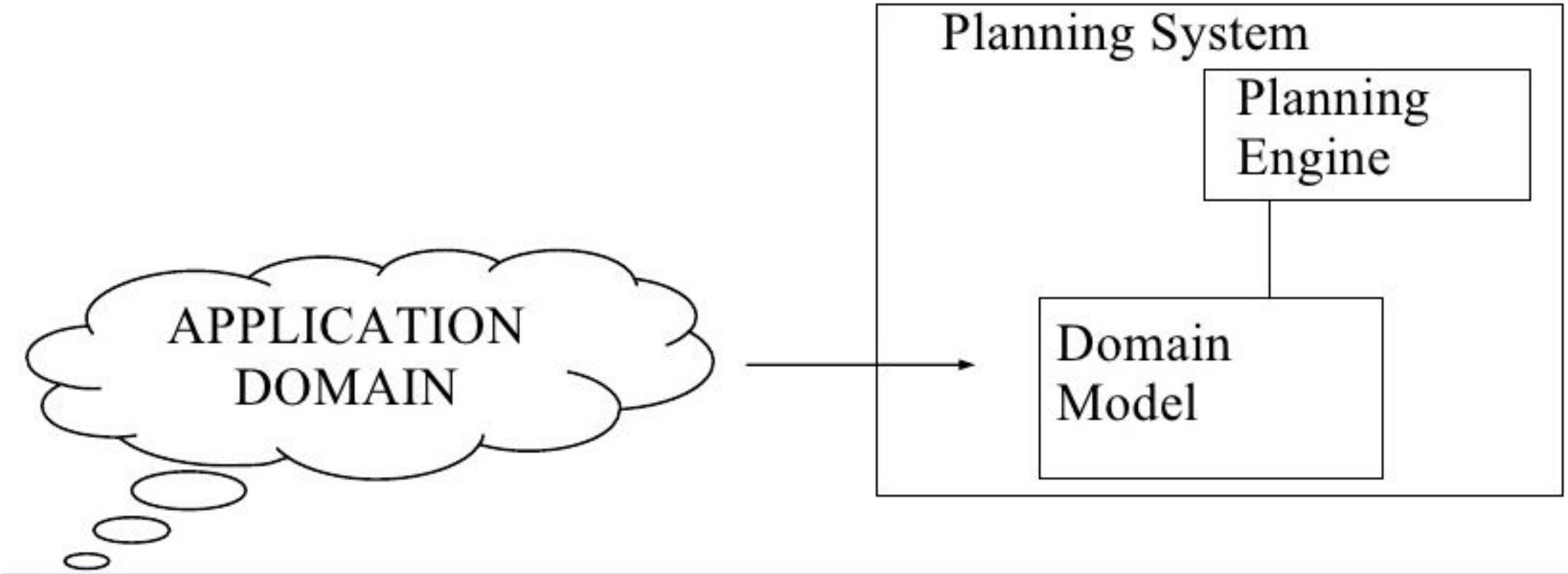
Knowledge Engineering for Automated Planning

What is knowledge engineering?

Main assumptions

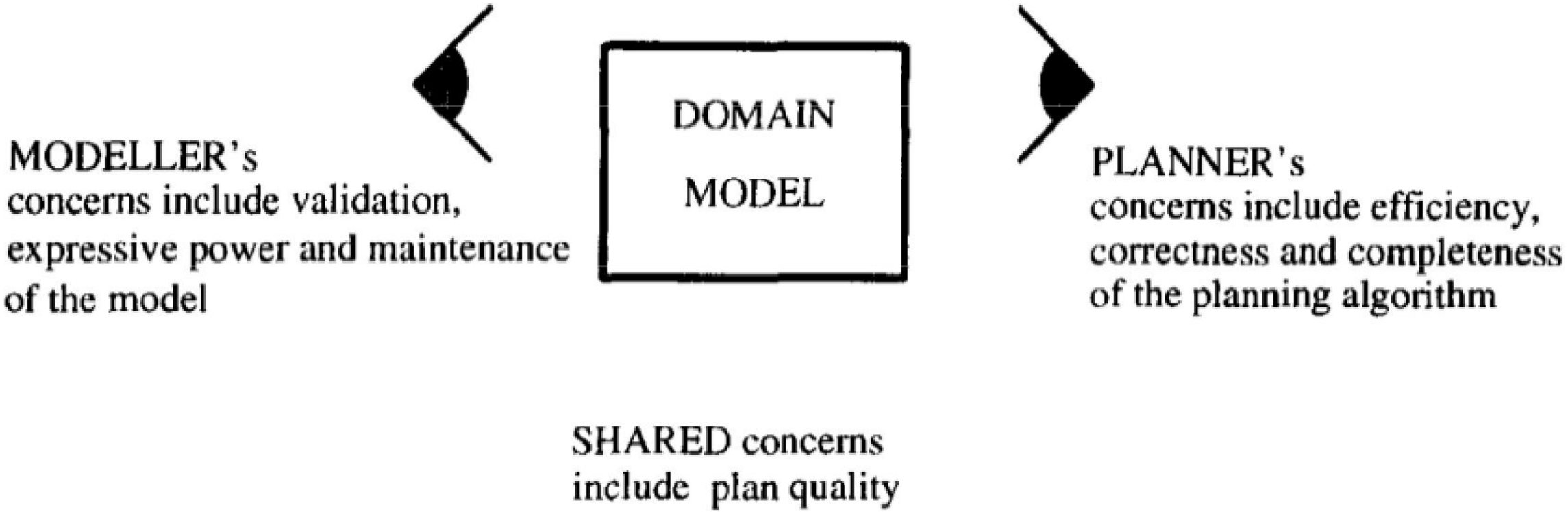


A main assumption in domain-independent planning is that there is a logical separation between the planning engine and the domain model

Implications

- **Positive**
  - planning engines AND domain models can be developed, tested, debugged, validated independently
  - domain models may be useful for more purposes than simply automated planning functions (e.g., validation, mining, simulation, etc.)
- **Negative**
  - Domain model representation is influenced by ‘what planners can handle’
  - Inefficiency in domain-independent planning systems on a specific domain of interest

Involved views on Model



T. McCluskey and J. Porteous. Engineering and compiling planning domain models to promote validity and efficiency. *Artificial Intelligence*, 95(1):1 – 65, 1997.

Knowledge Engineering

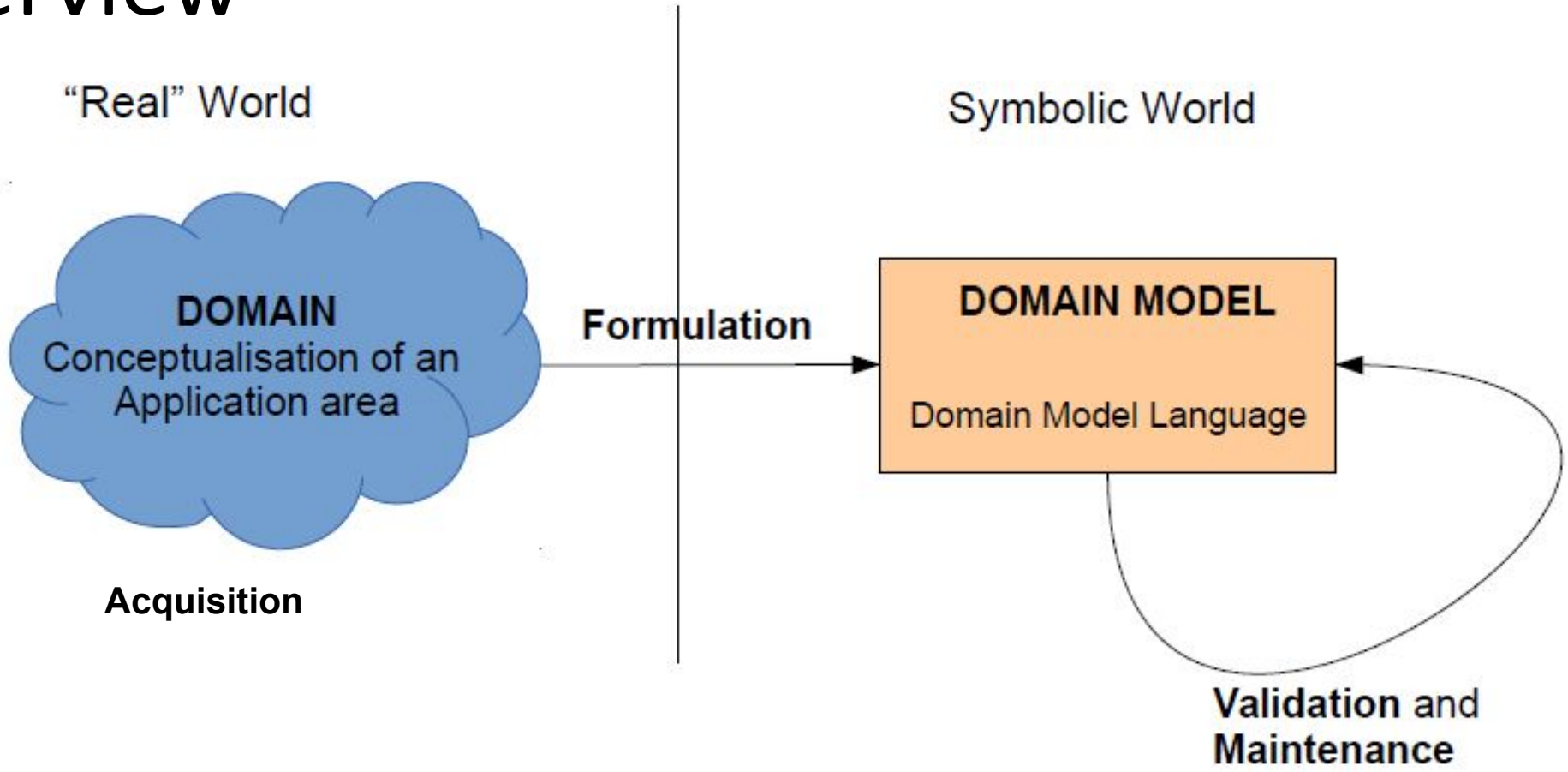
Knowledge Engineering for automated planning is the process that deals with:

- **acquisition**
- **formulation**
- **validation**
- **maintenance**

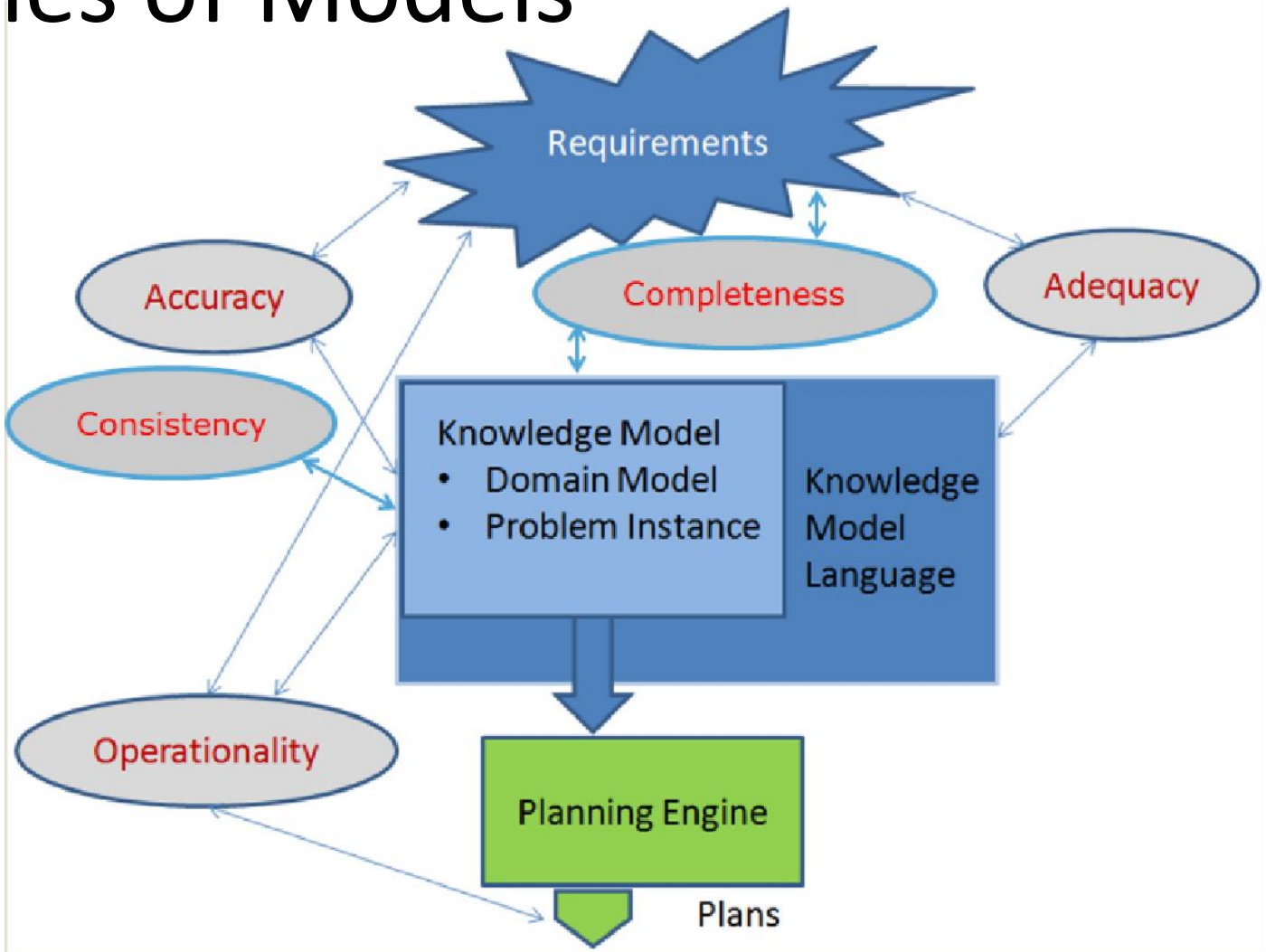
of planning domain models.

KE deals also with the selection and optimization of appropriate planning machinery to work on it.

Overview


"Real" World
Symbolic World
DOMAIN
Conceptualisation of an
Application area
Formulation
DOMAIN MODEL
Domain Model Language
Acquisition
Validation and
Maintenance

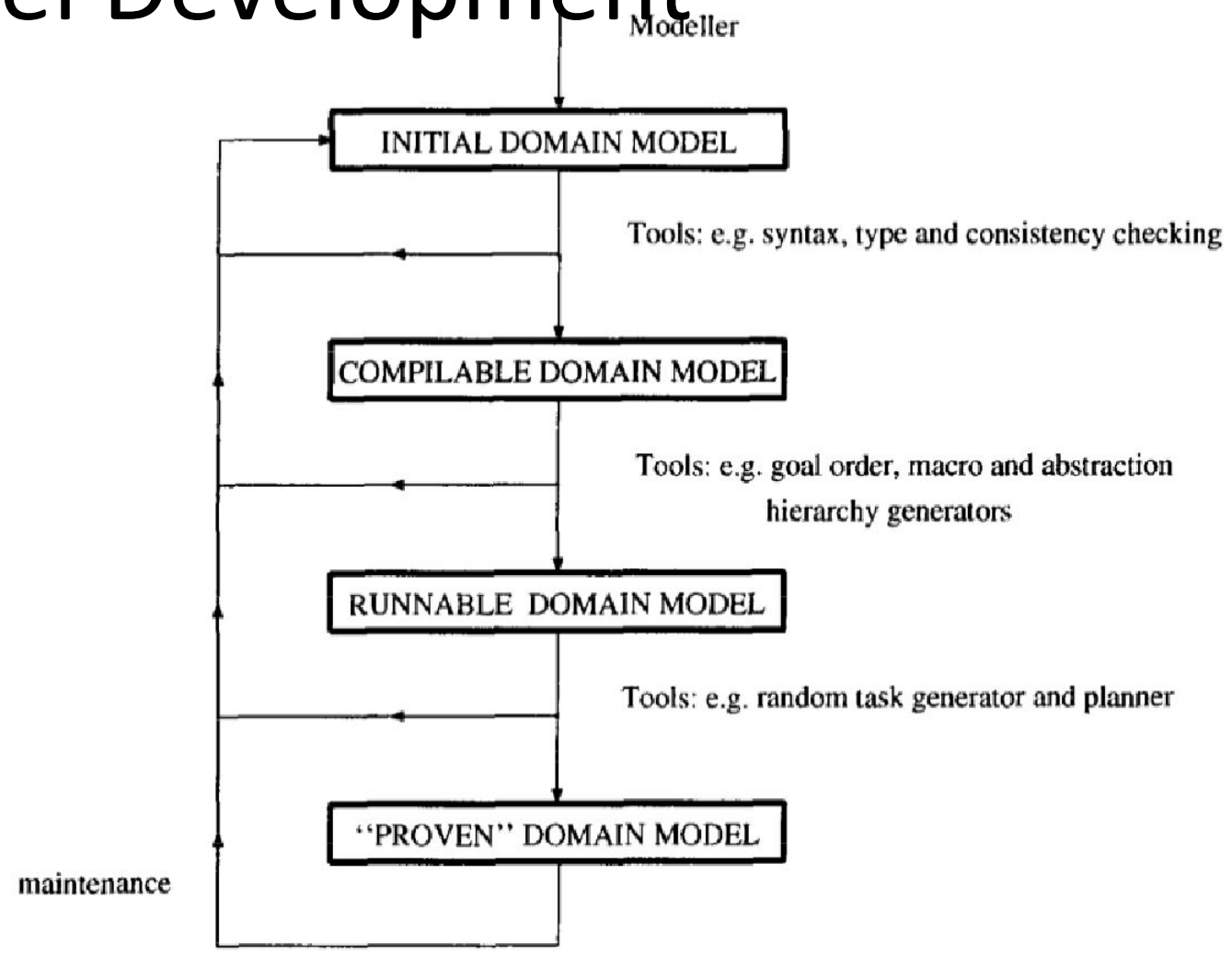
Properties of Models



Properties

- **Accuracy**: there exists a mapping between requirements and model's components.
- **Consistency**: a domain model is consistent if some interpretation exists that makes all its assertions true. (Subcase of the accuracy)
- **Completeness**: (*very* informally) any solution in the domain model is also a solution for the real-world domain, and vice-versa.
- **Adequacy**: the exploited language has the expressive power to represent the requirements (a model can be accurate, but not adequate).
- **Operationality**: there is a planning engine that can produce solutions within acceptable resource bounds.

The Model Development



Reformulation?

- Macro-operators
- Action schema splitting
- Domain model configuration
- Entanglements

Reformulation techniques for automated planning: a systematic review

ICKEPS 2024

ICKEPS Purposes

- to test how bright teams are
- to test competitors' knowledge and familiarity of PDDL
- to test the PDDL "hacking" skills of competitors

- effective ways of co-operating during the knowledge engineering process
- requirements for future supportive KE environments
- innovative ways of using tools to encode high quality models

Stages

- Modeling of a provided scenario
  (from the end of this presentation to 15.20)
- Submission
  15.20
- Modeling of second scenario and preparation of slides
  (15.20 to 17.30)
- Submission
  (17.30)

Modelling scenario (part 1)

- Let’s play a game…

Automated Planning plays a major role in driving AI players, and in automatising game testing!

FILUTA

What is needed for the submission?

First submission at 15.20

- The domain model and the models for each problem described in the scenario
- The logs of the planner used to generate solutions

Second submission at 17.30

- The new models (domain and problems)
- A diff between the domain models
- The logs of the planners used to generate solutions
- 3 slides where you briefly describe the model and the approach you implemented

Evaluation metrics

- KE process, team working, strategy, collaboration (slides)
- Quality of the models, correctness, readability, originality (models)
- Operationality of the models (planner’s logs)
- Extendability and generality of the models (diff file, models)

FAQ

- Can I change team?
  NO
- Can we use any version of PDDL?
  The most expressive language allowed is PDDL2.1 with numeric variables, but no time representation
- Can we use any planner we like?
  YES
- Is there any template for the slides to be submitted?
  NO
- Can we get an extension as my hamster eat my model?
  NO

Teams, Scenarios, and slides

Rule of thumbs

"*It is almost a law in PDDL planning that for every language feature one adds to a domain definition, the number of planners that can solve (or even parse) it, and the efficiency of those planners, falls* ***exponentially***"

(anonymous reviewer)

Links!

- ENHSP (planning engine)
- Metric-FF (planning engine)
- Fast Downward (a collection of planning engines)
- Unified Planning Framework (library of planning engines and parsers)
- Planning.domains (editor and online planning engine)
- FD online (editor and online planning engine)
- Lightweight Automated planning toolkit (engine)
- Planning wiki (well, it is a wiki!)
- PDDL plugin for visual studio
- Learn PDDL tutorial
- PDDL 2.1 description
- ICKEPS 2016
- https://www.diffchecker.com/text-compare/ (if you don’t have a diff tool on your laptop)

Some (useful?) references

- T. McCluskey and J. Porteous.
  Engineering and compiling planning domain models to promote validity and efficiency. *Artificial Intelligence*, 95(1):1 – 65, 1997.
- T. McCluskey, T. Vaquero, M. Vallati.
  Engineering Knowledge for Automated Planning: Towards a Notion of Quality. *K-CAP 2017*
- TS Vaquero, JR Silva, JC Beck.
  A brief review of tools and methods for knowledge engineering for planning & scheduling - *KEPS 2011*
- **The Fifth International Competition on Knowledge Engineering for Planning and Scheduling: Summary and Trends**