

Intelligent Monitoring of Software Components

Sabine Moisan

INRIA Sophia Antipolis Méditerranée, FR

Artificial Intelligence ←→ Software Engineering

Cross disciplinary approach in two directions

Software Engineering Components, frameworks,

Components, MDE

 Complement modelbased approach for deployment and run time adaptation of component processing chains models...

Knowledge-based systems Supervision techniques

 Develop adaptable toolkits to design and customize KBS elements (GUIs, languages, KB verification tools, inference engines ...)

Knowledge bases,

supervision rules...

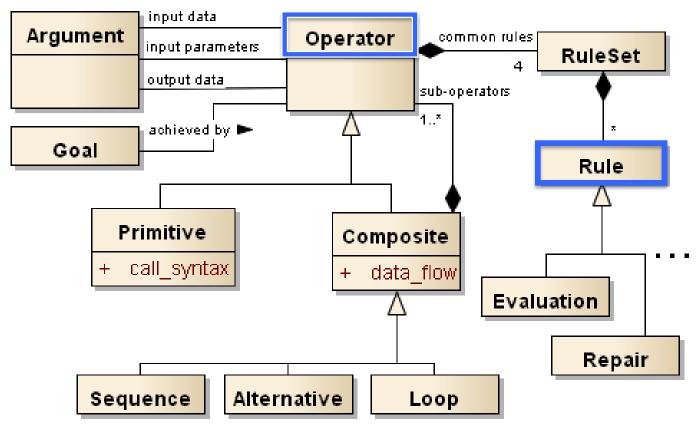
Artificial Intelligence

Supervision for (dynamic) software component adaptation

Knowledge-based approach to component configuration (select, assemble, tune...) at deployment and at run time

- → At runtime: Changing environment
 - Control execution of running components
 - Adapt configuration in case of changes
 - Handle failures: detect and fix
- Provide maximum autonomy (embedded systems)
 Supervision => means to express and operationalize this knowledge

Supervision Ontology

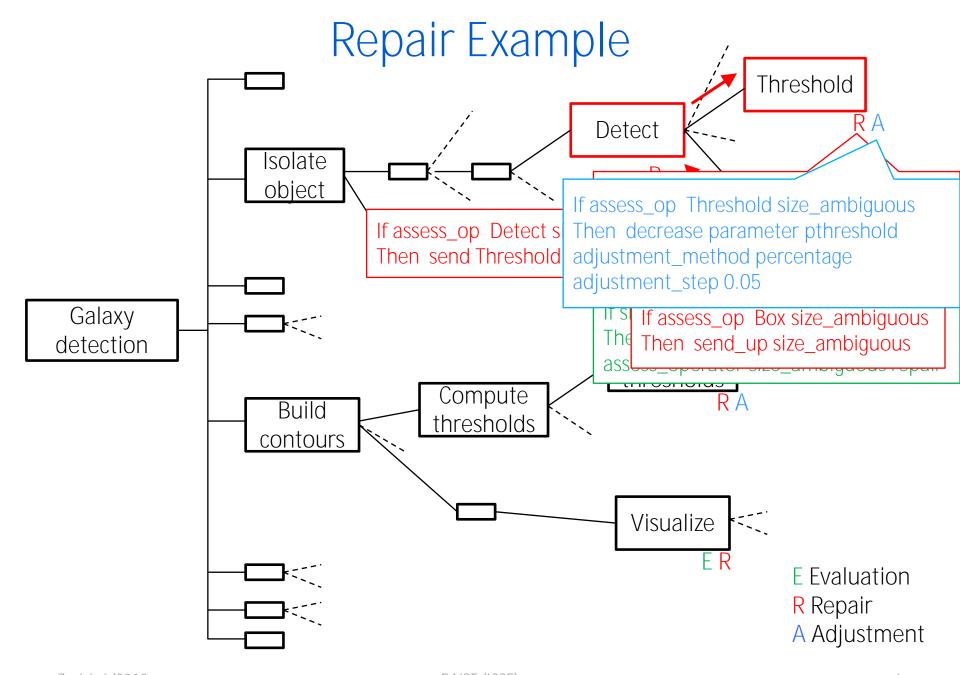


- Generic concepts & relations necessary to select, assemble, monitor, control... components
- Knowledge Representation Language

Policy Rules - Evaluation & Repair

- Know-how to take (run time) decisions
 - Involves different categories of rules
 - Translated into executable code: part of the run time system
- For dynamic adaptation
 - Evaluation: assess results, detect problems
 - Repair/adjustment: propagate problems, fix problems (tune parameters, reorganize configuration plan)

Choice	If object attribute a has value v Then use operator op ₁
Initialization	If object attribute a has value v Then set parameter p to value v ₁
Evaluation	If result r has property p //detected automatically or manually Then declare problem pb for op ₁ and call repair
Repair/ Adjustment	If operator op ₁ has problem pb Then transmit pb to op ₂ or increase/decrease parameter p ₁ or choose another sub-operator op ₃ for future execution



Conclusion & Future Work

Used in Al applications

- Customizable mechanism
- Explicit way to express evaluation/repair policies
- Performance: Limited overhead

But... evaluation/repair rules and policies difficult to specify... and to automate

Now: Complement more formal/but limited methods (logical constraints)

Models@Runtime + Supervision

Change Model transformation Application Model (requirements, context...) Implementation Model (build configurations...) Logical constraints Introduce evaluation/repair Sub-model ... most often! supervision mechanism to (set of configurations) guide exploration of search + Supervision space and fine tune solution Configuration

Any question?

