Model-Based Stochastic Analysis with Probabilistic Graph Query Evaluation

Máté Földiák, Kristóf Marussy, Dániel Varró

Context

Critical Cyber-Physical System

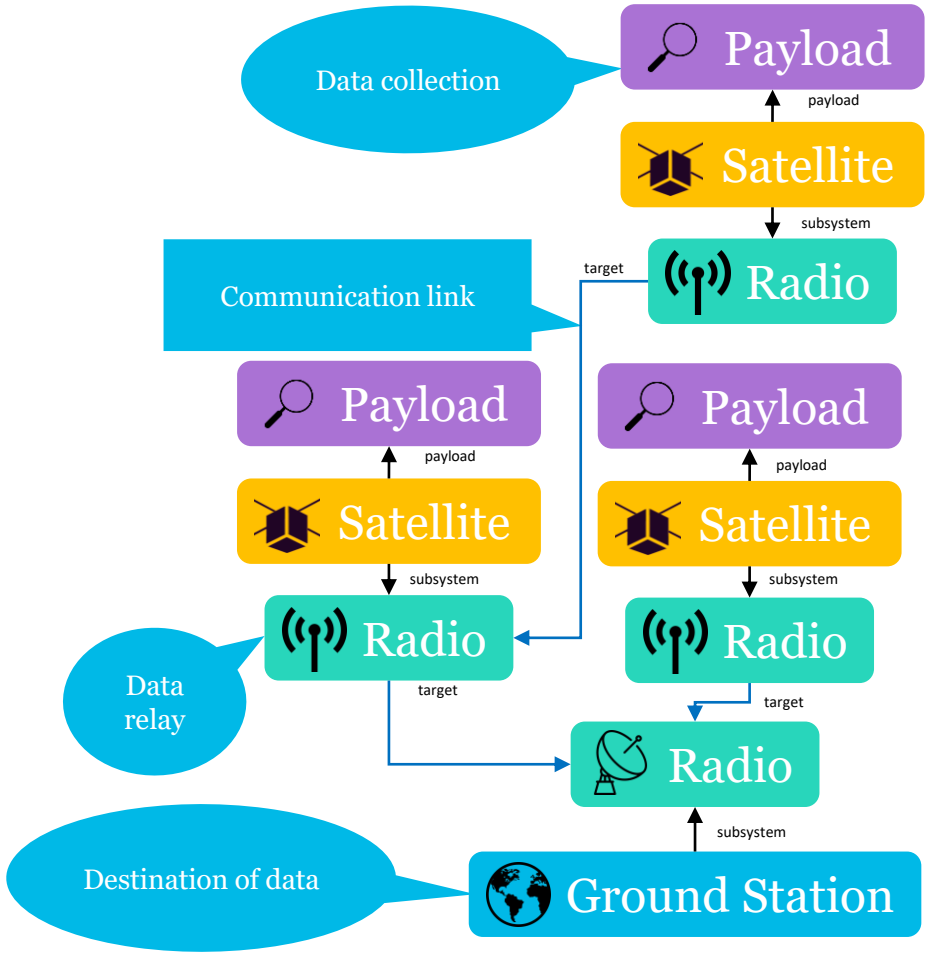
- Correct operation must be ensured
  - Formally verified or
  - Systematically tested
- Stochastic failures are present
  - Component degradation
  - Environmental conditions
- Conflicting objectives
  - Safety, reliability, cost
  - Optimization, design decisions
  - Influenced by early design

Early System Design

- Requirements, constraints
- High level concepts related to the system
  - Abstraction with a system architecture model
- Previous experience
- Unknowns in
  - System architecture
  - Failure management, reliability
  - Behaviour
- Frequent changes

Running Example: Architecture Models

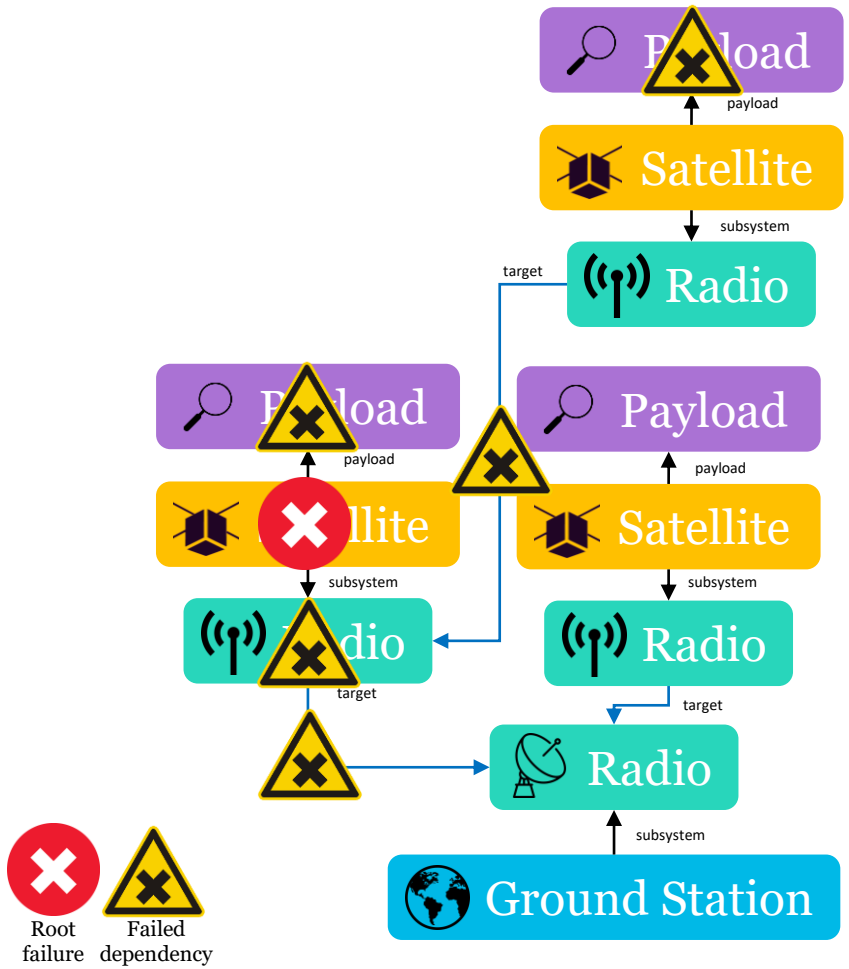
- Satellite constellation to measure background radiation
  - Performance measure
- Design system architecture
  - What components to use
  - Arrange communication topology



S. J. I. Herzig, S. Mandutianu, H. Kim, S. Hernandez and T. Imken, "Model-transformation-based computational design synthesis for mission architecture optimization," *2017 IEEE Aerospace Conference*, USA, 2017

Running Example: Reliability Modeling

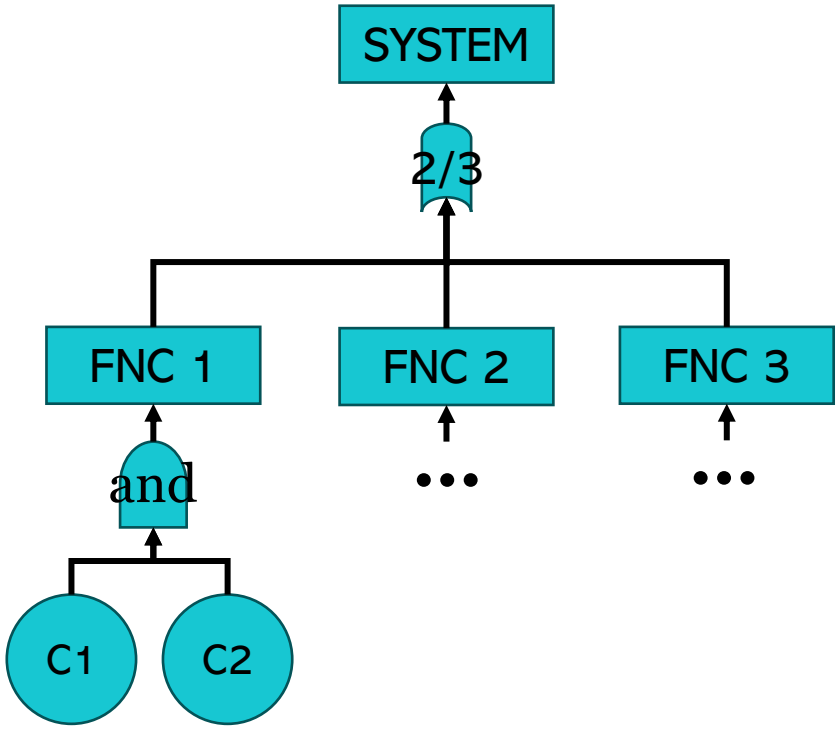
- Satellite constellation to measure background radiation
  - Performance measure
- Design system architecture
  - What components to use
  - Arrange communication topology
- Component level failures
  - Reduced performance
- Optimize for expected performance
  - Performability



Background

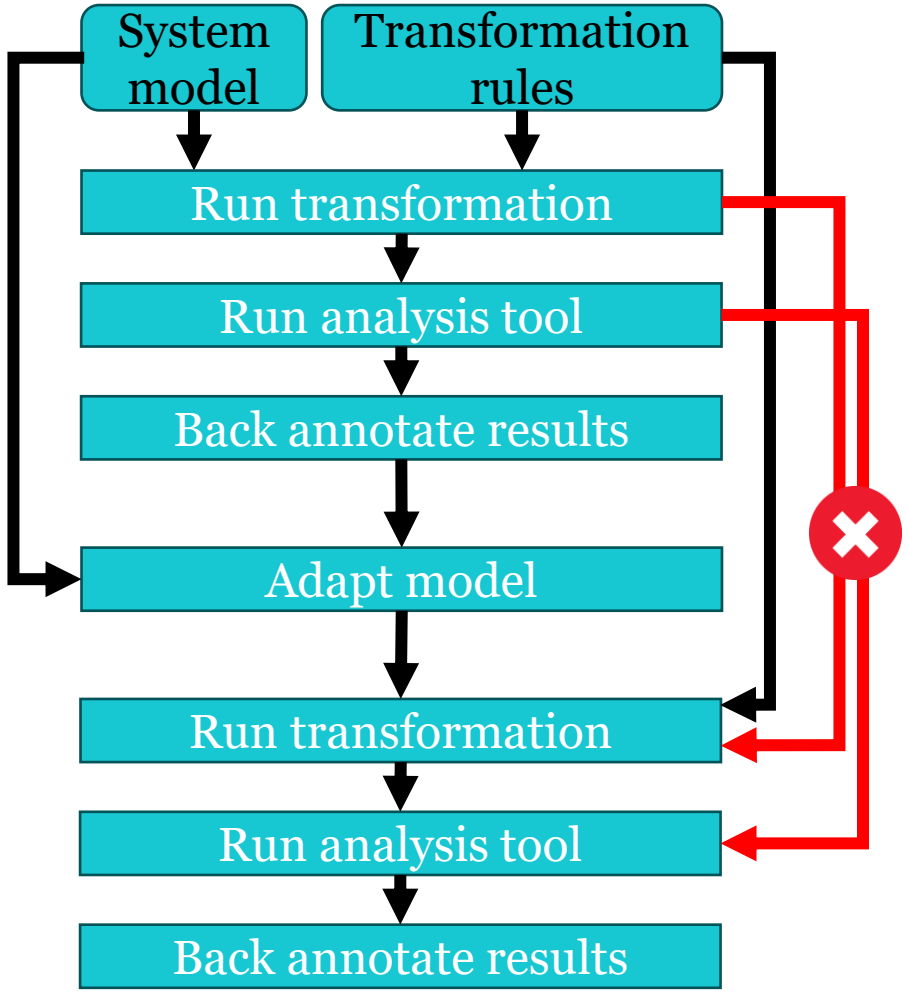
Performability Analysis

- Expected performance in presence of failures
- Basic event
  - Atomic, independent failures
  - Failure probability is known
- Compound events
  - Corresponds to higher level functionality
  - Depends on other events
- Performability
  - Indicator of the architecture quality



Traditional Analysis Toolchain

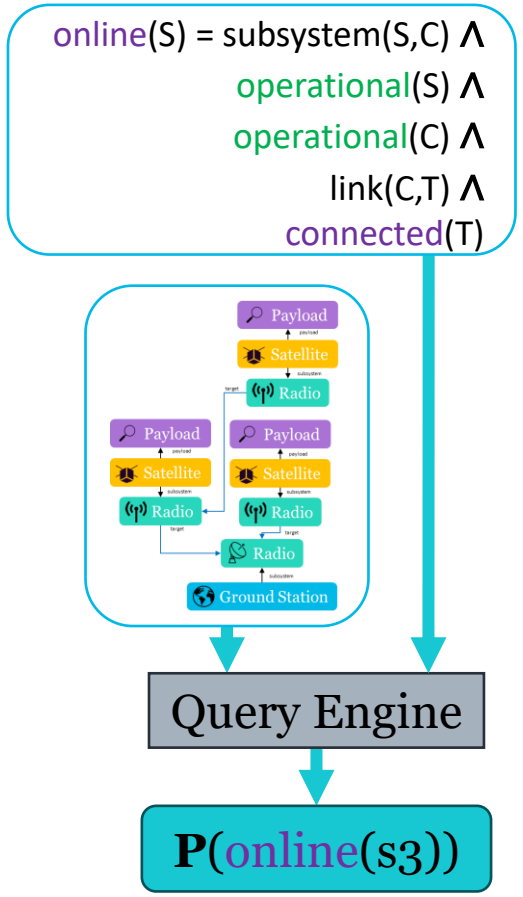
- System model in SysML, EMF, etc.,
- Transformation rules
  - From modelling language to stochastic analysis language
  - Epsilon, Xtend, etc.
- External analysis
  - Fault trees, Markov chains, Petri nets
  - Analysis tools are separate from modeling tools



Lifting Stochastic Analysis to Model Level

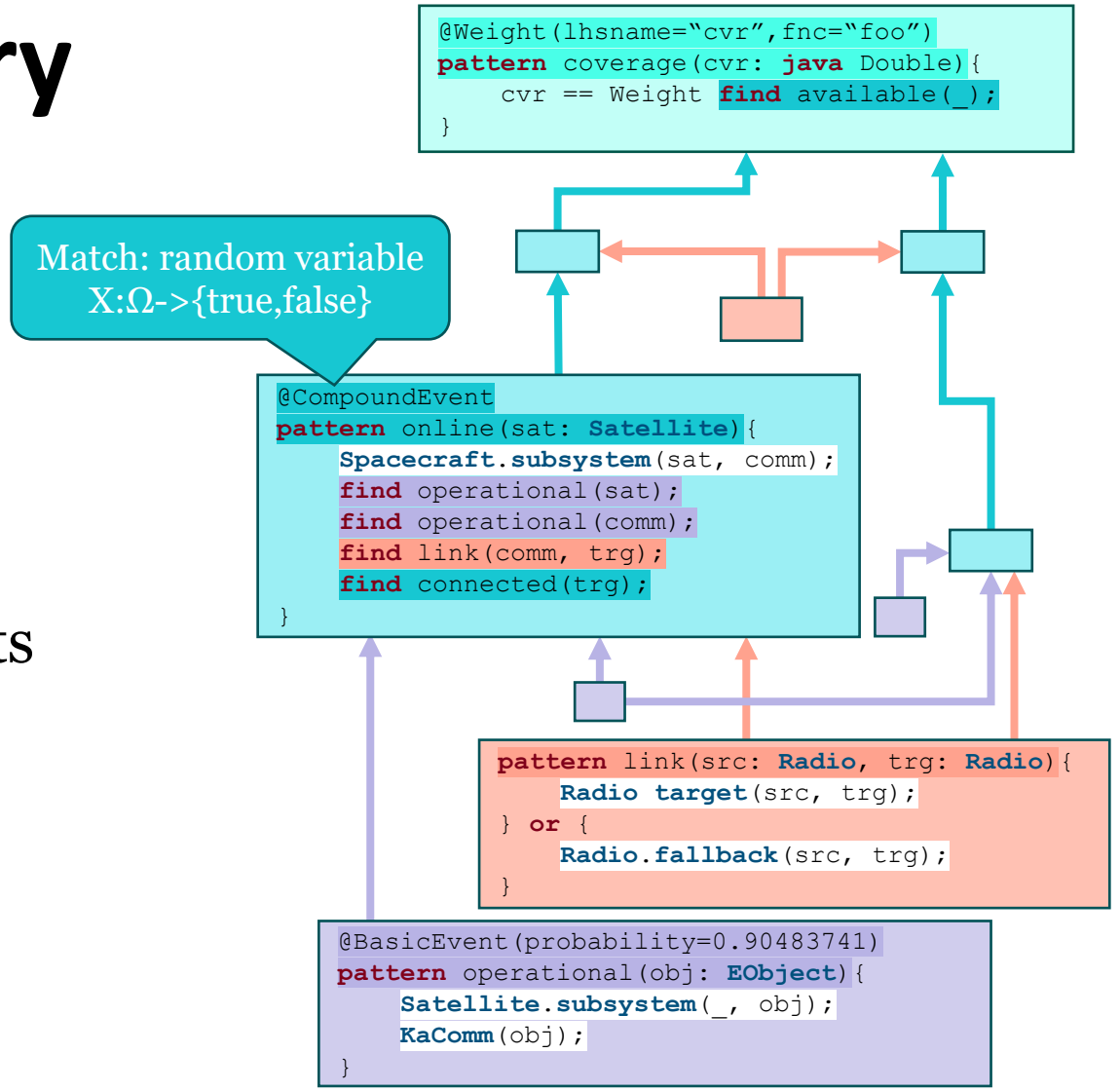
Lifting Stochastic Analysis to Model Level

- Challenge: Create an analysis method that is
  - High level, and scalable (in model size)
  - Support incremental changes
- Core ideas
  - Reuse graph predicates for stochastic analysis
  - Leverage query engines for efficient reevaluation
- Expected outcomes
  - Model-based formal analysis
  - Reduced engineering complexity



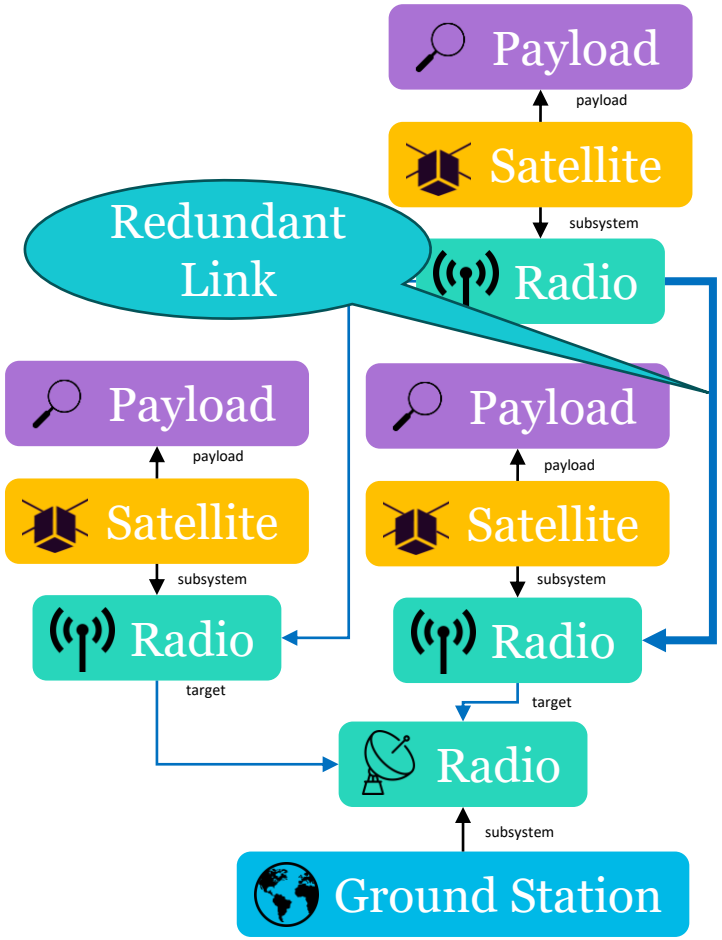
Probabilistic Graph Query

- Event semantics (instead of Boolean)
  - Based on discrete probability theory
  - Declarative failure modeling
- Lightweight language extension
  - Distinguished queries for basic events
    - component state (*operational*)
  - Override operator semantics
  - Evaluate match probability
    - Expected value



Incremental analysis

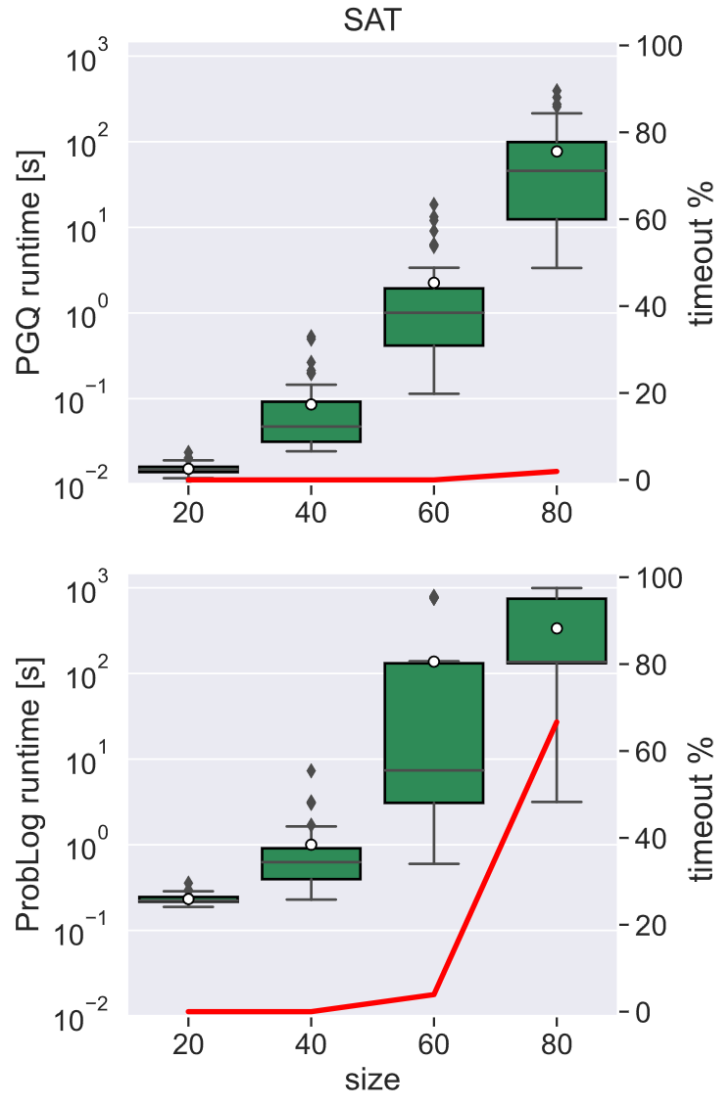
- Small changes in the system model
  - Made by engineers, tools, AI
  - Not affecting the whole model*
- Incremental analysis
  - Retain intermediate calculations results
  - Detect and propagate changes
    - What is changed?
    - What is affected?
- Inherited from underlying technologies



Evaluation

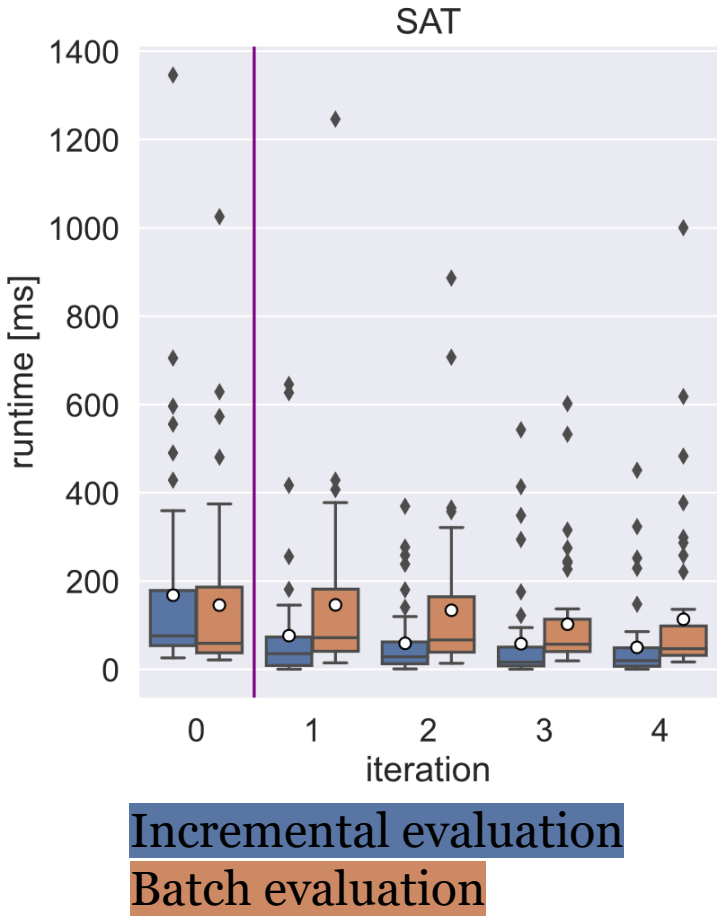
Analysis performance

- Evaluate 30 models with different tools
  - Measure runtime
    - Completed analysis only
  - Time limit of 20 minutes
    - % of failed runs
- Baseline: ProbLog
  - Similar formalism to PGQ
- Reduced analysis time



Incremental Analysis Performance

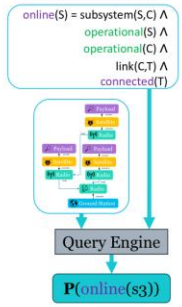
- Apply realistic changes, and reevalute
  - E.g., add a new component
- Baseline: batch analysis with PGQ
- Results
  - ~50% reduced mean analysis time
  - Dependent on domain, and change



Conclusions

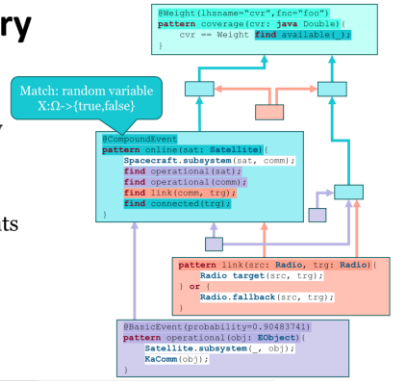
Lifting Stochastic Analysis to Model Level

- Challenge: Create an analysis method that is
  - High level, and scalable (in model size)
  - Support incremental changes
- Core ideas
  - Reuse graph predicates for stochastic analysis
  - Leverage query engines for efficient reevaluation
- Expected outcomes
  - Model-based formal analysis
  - Reduced engineering complexity



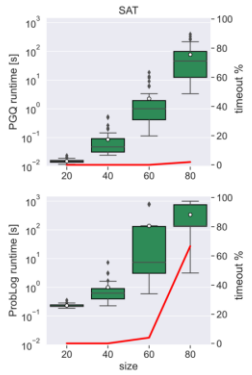
Probabilistic Graph Query

- Event semantics (instead of Boolean)
  - Based on discrete probability theory
  - Declarative failure modeling
- Lightweight language extension
  - Distinguished queries for basic events
    - component state (*operational*)
  - Override operator semantics
  - Evaluate match probability
    - Expected value



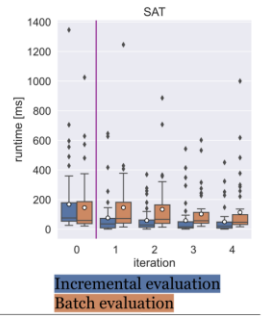
Analysis performance

- Evaluate 30 models with different tools
  - Measure runtime
    - Completed analysis only
  - Time limit of 20 minutes
    - % of failed runs
- Baseline: ProbLog
  - Similar formalism to PGQ
- Reduced analysis time



Incremental Analysis Performance

- Apply realistic changes, and reevalute
  - E.g., add a new component
- Baseline: batch analysis with PGQ
- Results
  - ~50% reduced mean analysis time
  - Dependent on domain, and change



LINKÖPING UNIVERSITY WASP WALLENBERG AI, AUTONOMOUS SYSTEMS AND SOFTWARE PROGRAM ftsrg MŰEGYETEM 1782