Efficient Model Transformation Languages

Cláudio Gomes
TrNet: A Language and a Workbench for Efficient Model Transformations

- Efficient declarative model transformation languages
  - Explicitly model the pattern matching process
  - Develop analyses and optimizations
TrNet: Syntax & Semantics

• Network of sets of pattern instances

• Operators combine source sets to create new sets subject to semantic conditions and functions
Typical Transformation Process
Analyses

- Input models statistics
- TrNet model

Analysis

Estimated number of instances in each set, i.e., approx. size of output model
Optimizations
Case Study: DSLTrans Compilation
Evaluation

Activity Migration (from DSLTrans)

Depth

Pattern and Operator Reduction with OPM

Number Of Patterns
Number Of Operators

Transformation Times

Transformation Time (ms)

Model Size (N)
Conclusions

• Mitigate performance impact of high level languages
  – ++Productivity
  – ++Performance

• Domain specific optimizations can be applied as HOT*
Current & Future Work

- Co-simulation of FMUs
- Correct co-simulation
  - Absence of deadlock
  - State events are detected
  - Correct synchronization between interfaces
  - Etc...
- Efficient co-simulation
Thank you!