Massif - the love child of Matlab Simulink and Eclipse

Ákos Horváth, István Ráth and Rodrigo Rizzi Starr
(ahorvath@mit.bme.hu)
Outline of the talk

- Motivation and background
  - Integration with Matlab Simulink

- Massif
  - Overview
  - Architecture

- Feature set
  - Import/export
  - OSLC interface

- Conclusion
  - Current status
  - Future work

- Main Contributors
  - Ábel Hegedüs
  - Rodrigo Rizzi
  - Mártin Búr
  - Lincoln Nascimento
  - Róbert Dóczi
  - Samoel Mirachi
  - István Ráth
  - Ákos Horváth
Background: Matlab Simulink

- **Matlab**: multi-paradigm numerical computation
  - matrix operations, algorithm development, data visualization etc.

- **Simulink**: dataflow modeling environment
  - Modeling and simulating dynamic systems
  - Block diagram based

- Large user base > 1M
  - Automotive
  - Avionics
  - Railway
  - Etc.
Motivation: yet another tool integration

- Started in 2011 (as a second iteration)
  - Provide HW-SW allocation for early simulation of avionics systems
  - **Input**: Low-level Simulink libraries
  - **Output**: Integrated HW-SW Simulink model

- Allocation defined on high abstraction level → Eclipse
  - Hierarchical graph based visualizations → *yFiles + EMF-IncQuery Viewers*
  - On-the-fly model validation → *EMF-IncQuery*
  - Generic bridge between Matlab Simulink and Eclipse → *Massif*
Massif

Integrating Eclipse and Matlab Simulink
Massif – Eclipse-Matlab Simulink bridge

- **Generic Bridge between Matlab Simulink and Eclipse**
  - Import/Export Matlab Simulink systems and libraries
  - Traceability for incremental processing
  - Multiple import options for different usage scenarios
    - e.g., analysis or allocation

- **User-friendly EMF representation**
  - Manipulating systems/libraries directly on EMF level
  - Support for easier navigability

- **Works with Matlab license server**
  - Can connect to a running Matlab instance

- **Pure Java realization, EPL license**
  - Project specific constraint!
Massif - architecture overview

- Communicating with Matlab
  - Java RMI based communication
  - Server initialized within Matlab
    - Java Matlab Interface
  - Works with license server
- Generic EMF metamodel
  - EMF level operations translated into Matlab commands
  - Same library mechanism as in Matlab Simulink
  - FQN based traceability
Feature set

What Massif can do
Import/export models

- Tool configuration and initialization
- Import and export of Matlab Simulink systems
  - Import strategies
- Adding/deleting an element to/from the EMF model
  - Incremental refreshing
Shallow import option

- Only blocks within non-referred systems are imported hierarchically
Deep import option

- Each block inside each subsystem is imported. Each referenced model is imported as an individual model with direct model referencing in the parent model.
Each model reference block is imported as though it was a subsystem.
Flattening import option

- Each model reference block is imported as though it was a subsystem
Referencing import option

- For blocks with active library links, each source library is imported once as an individual model (but may be referenced multiple times)
Additional Features

- **Import Filters**
  - For leaving out unwanted elements/details from imported models

- **Bus creator and selector configuration**
  - Configured via string (fqn like) in Matlab Simulink
  - Use EMF EReferences to define and Massif automatically calculates configuration

- **Easier Block-to-Block navigability**
  - Derived EReferences defined by EMF-IncQuery
- **Open Services for Lifecycle Collaboration (OSLC)**
  - API for tool integration

- **Live OSLC API for Matlab Simulink**
  - Dynamically executes queries directly on the Matlab side
  - Uses the same Command Evaluator Server
  - Based on **Eclipse Lyo**
Conclusions

What we have done and where to move forward
Conclusions

- **Initial version is out**
  - Basic import/export features supported
  - EMF level handling of Simulink models

- **Experience**
  - **Avionics**
    - Largest model took ~2 hours 😞
  - **Automotive**
    - 5k elements within minutes 😊
  - Typical application scenarios
    - Offline processing of EMF representations
    - Library import → instance model generation on EMF side
Final points

- The examples and more details are available from:
  - https://github.com/FTSRG/massif

- Version 0.4 is available:
  - http://incquery.net/update/massif/release/site/

- Contributors:
  - Main: BME-FTSRG, Embraer S.A.
  - Auxiliary: IncQuery Labs Ltd.
  - Supporting projects: Trans-IMA, Concerto (EU-Artemis)

- Your contributions (feedback, forum posts, ideas, bugzillas, patches) are very welcome!
  - To what direction should we enhance Massif?