Harokopio University of Athens

Enabling System Models Automated Evaluation through Cross-Concept Information Utilization

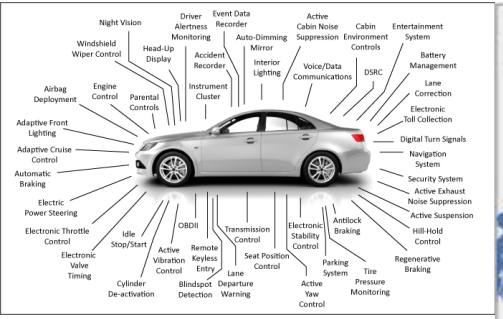
George-Dimitrios Kapos

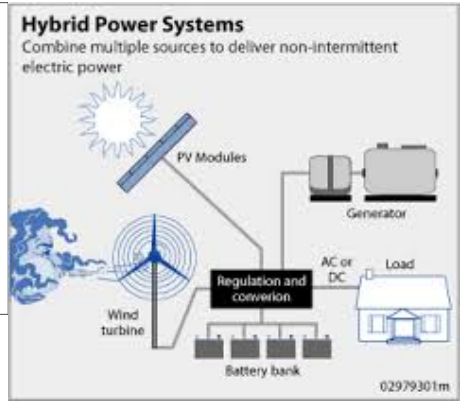
IEEE RCIS 2015 9th International Conference on Research Challenges in Information Science, May 13-15 2015, Athens, Greece

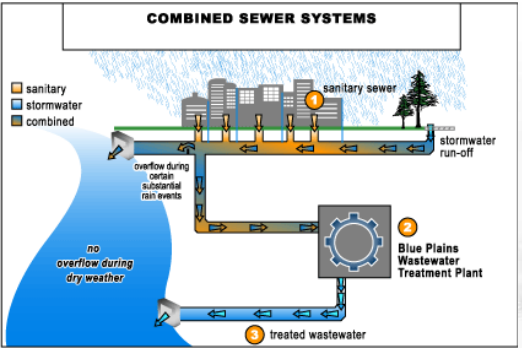
Outline

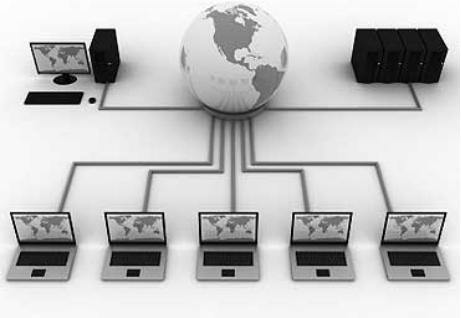
- Introduction
- Motivation
- Aim - Objectives
- Proposed Methodology & Framework Specs
- Application with DEVS
- Contribution
- Further & Future Work

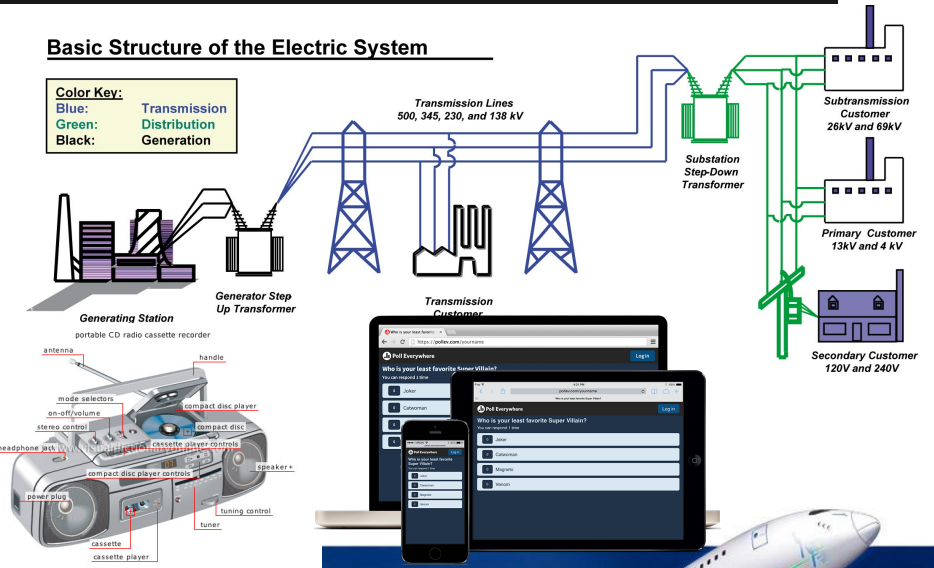
Systems are All Around Us





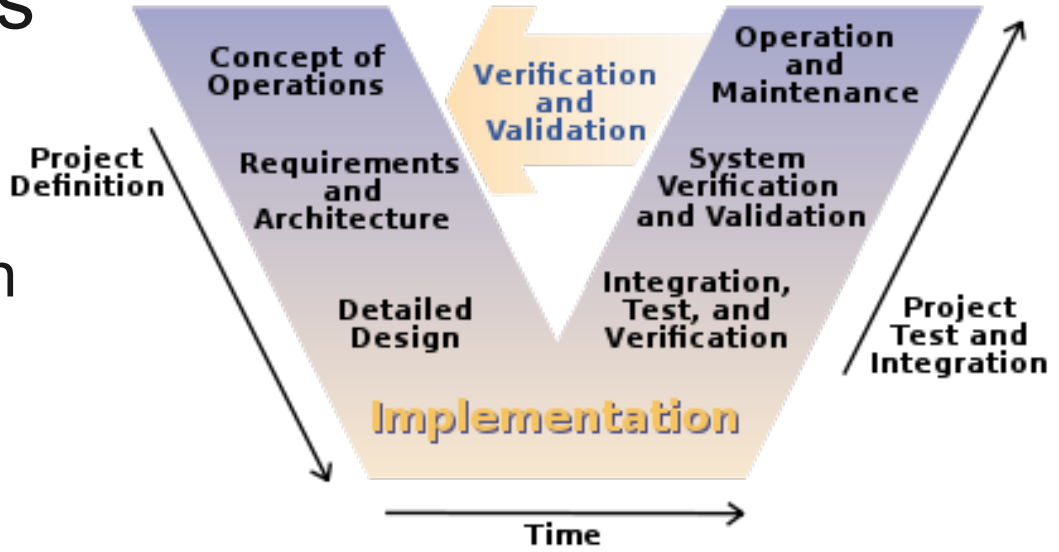






Systems Development Process

- Development Process of a Single System
  - Diverse activities
  - Many stakeholders with different concerns
  - Different tools support different activities
  - Different ways to specify system information



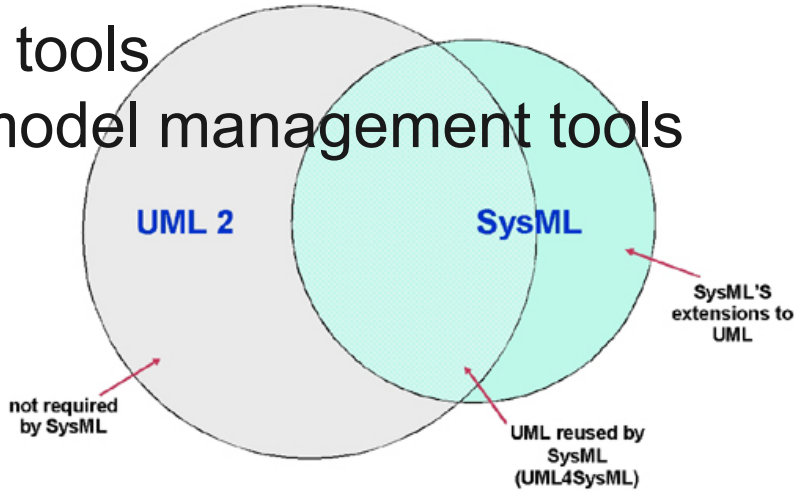
The V-model of the systems engineering process, from Clarus Concept of Operations. Publication No. FHWA-JPO-05-072, Federal Highway Administration (FHWA), 2005

Model-Based System Engineering

- Model-Based System Engineering (MBSE)
  - A single model throughout the whole process
  - Diverse pieces of information (requirements and specifications) are combined as distinct views
  - Seamless transition to the next phase
  - Lowers cost of changes to preceding activities
  - A comprehensive perspective of the system is developed and is available at any time

Systems Modeling Language

- Systems Modeling Language (SysML)
  - OMG standard
  - A UML profile
  - Supported by UML modeling tools
  - Usable in standards-based model management tools



OBJECT MANAGEMENT GROUP

SysML

- With SysML
  - Information regarding the systems is defined in an open and standards-based manner
  - But some activities are not supported inherently, e.g. **simulation** of system models

Motivation

- Simulation of system models is **crucial**
  - It may provide information that would be conceived several phases later
  - Requirements verification can be facilitated
  - Huge amounts of resources can be reserved compared to discovering system deficiencies later
- Valuable information in the system model:
  - must be located, complemented with additional info (e.g. performance attributes) and transformed

Motivation

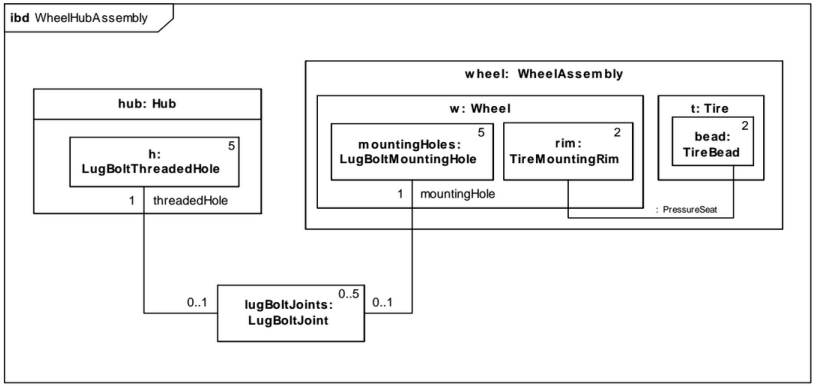
- Simulation of SysML models has been addressed, but:
  - Generation of simulation code is not fully automated
  - There is dependence on proprietary solutions
  - They are applicable in specific domains
  - They are coupled with specific simulators
  - Simulation results are not available in the system model

Related Work

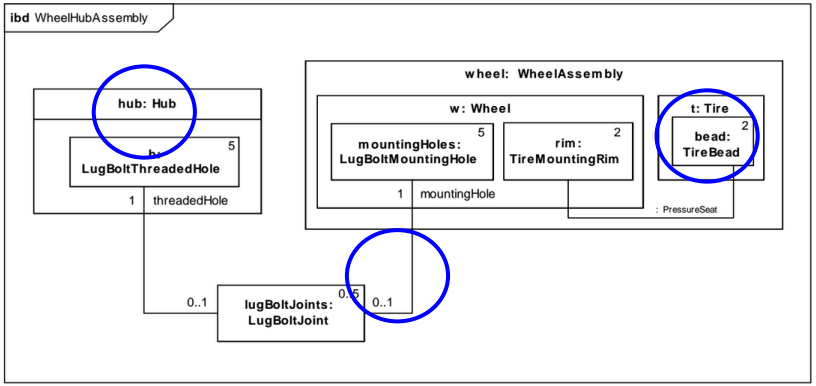
Table 1: Criteria satisfaction (full ✔, partial ❍ or no ✗) of related work

Aspect	CASSI	SLIM	MARTE	TTool	SysML4Modelica
Systems Oriented (SysML)	✔	✔	❍	✔	✔
Open (Non proprietary)	✗	✗	✔	❍	✔
Domain Independent	✗	✔	✗	✗	✔
Simulator Independent	✗	✔	✗	✗	✗
Automated Simulation Execution	❍	✗	✗	✔	✔
Incorporation of simulation results	✗	✗	✗	✗	✗

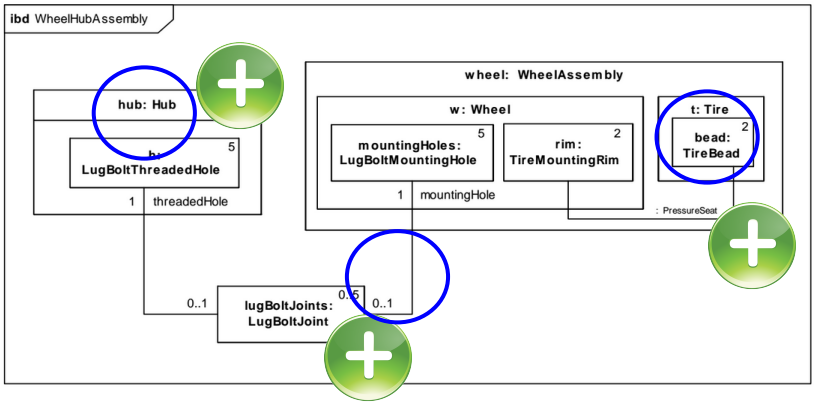
Utilize the System Model



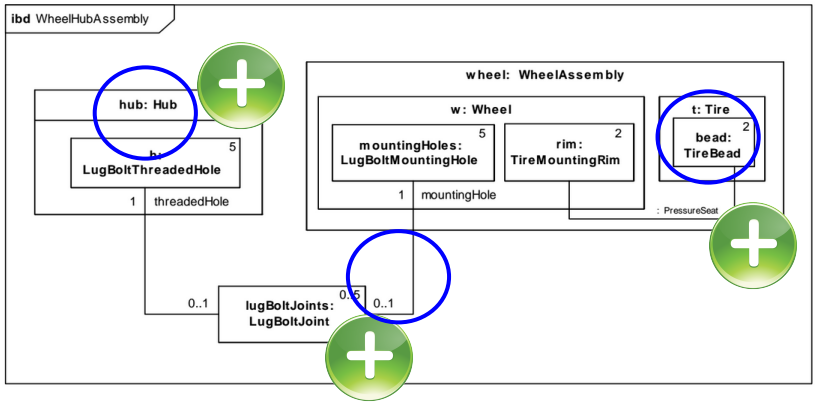
Identify Useful Information

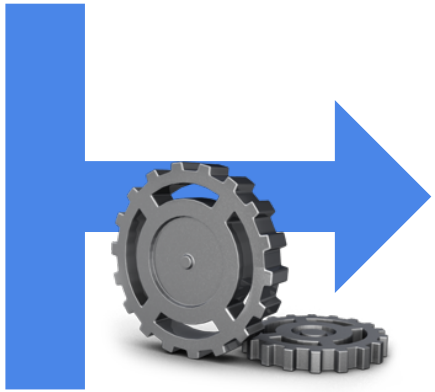


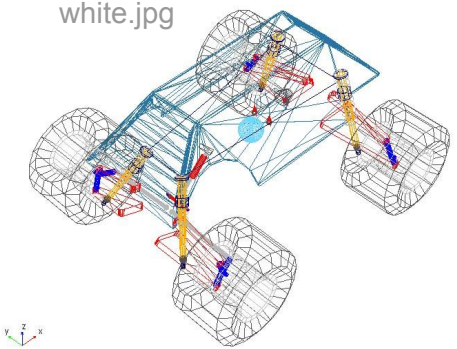
Add Behavioral Information



Combine to Form Simulation Model

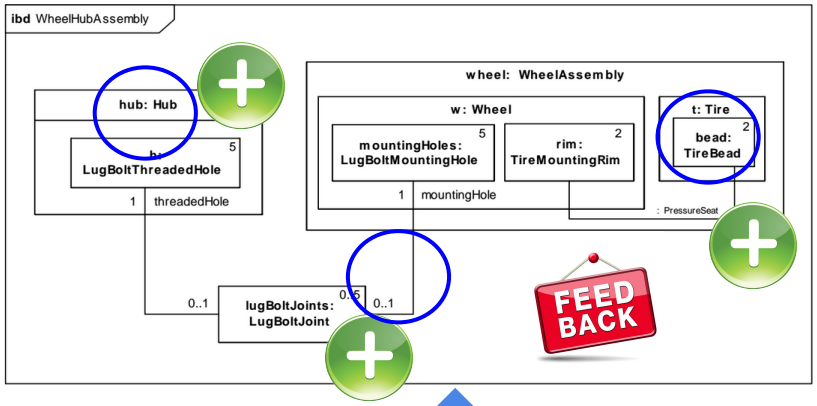


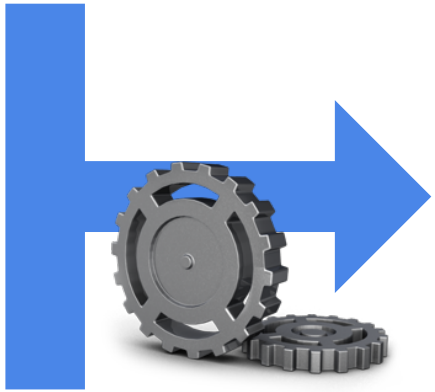




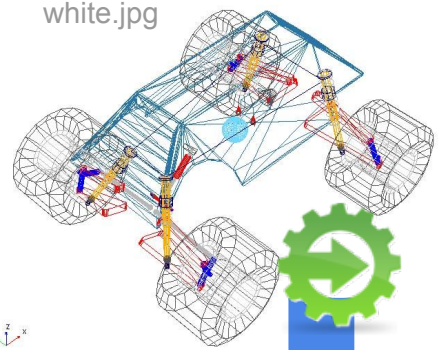
http://robots.mit.edu/projects/old_projects/darpa/images/front_iso_wireframe_white.jpg

Simulation Results Feedback

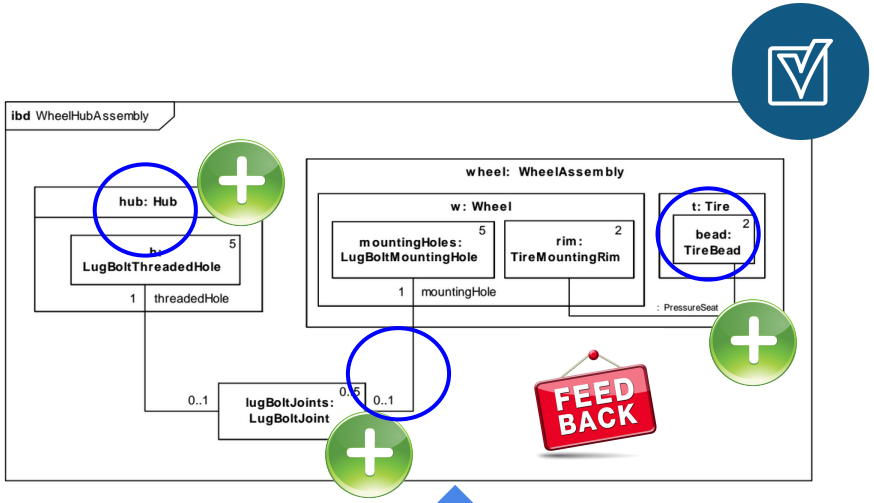


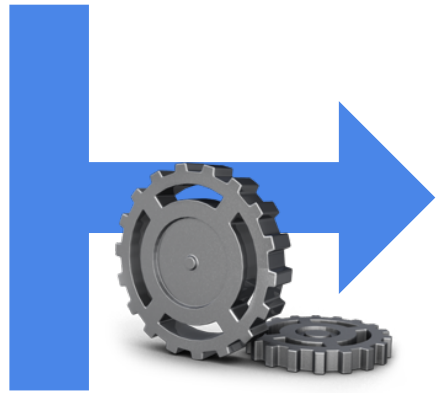


http://robots.mit.edu/projects/old_projects/darpa/images/front_iso_wireframe_white.jpg

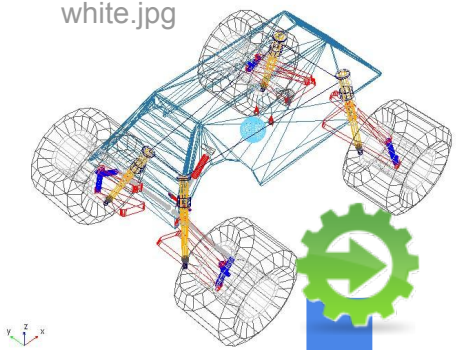


Enable Automated Requirements Verification (System Model)

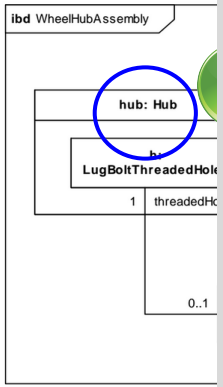




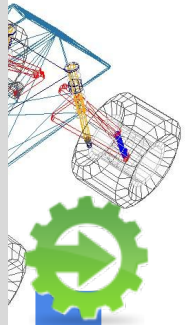
http://robots.mit.edu/projects/old_projects/darpa/images/front_iso_wireframe_white.jpg



Principles that Add Value



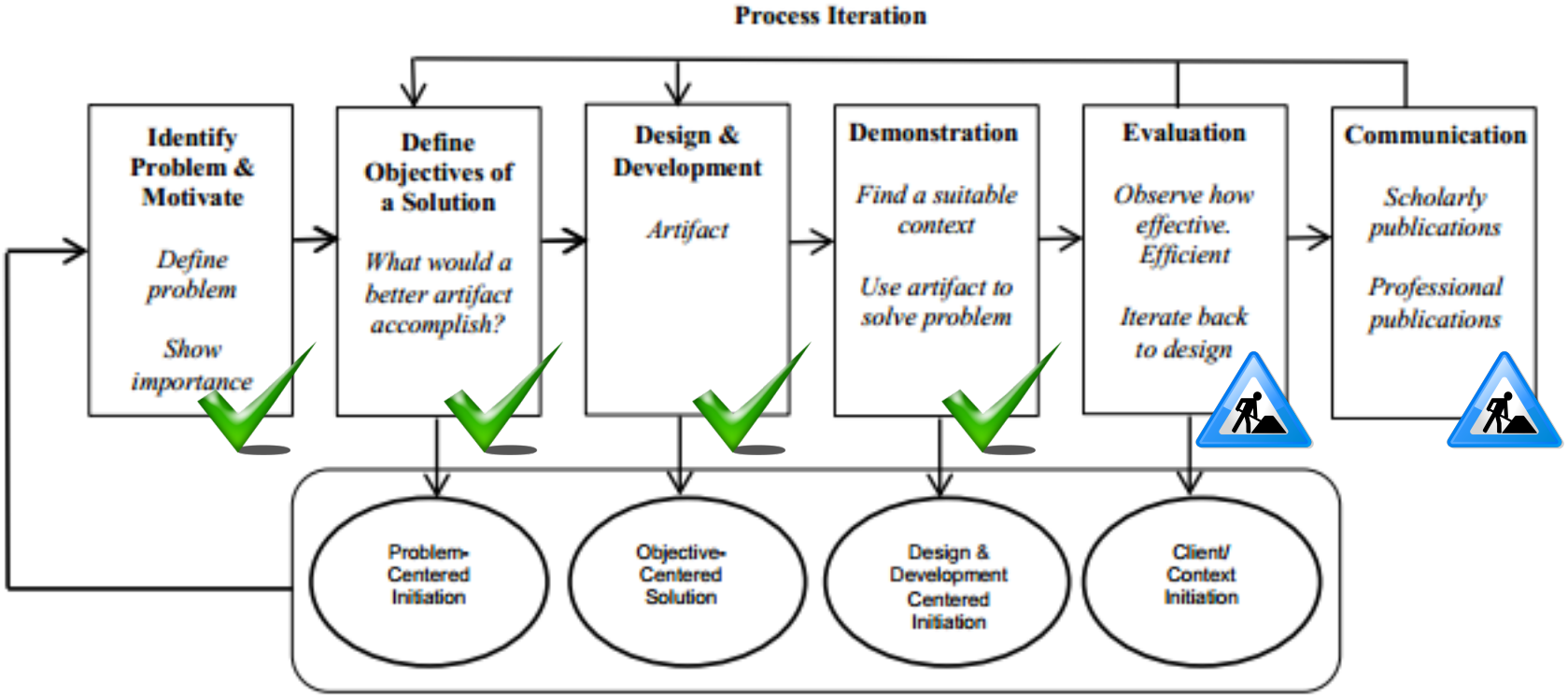
- General approach
- Open representation for information
- Standards-based
- Domain independent
- Simulator independent
- Promote interoperability
- Full automation



Aim

- Enhance system model evaluation by:
  - proposing a methodology for automated simulation of SysML models that can be applied across different domains
  - specifying corresponding implementation frameworks that may deliver such enhancements imposing minimum restrictions (specific tools & solutions)

Research Methodology: Design Science



Design Science Research Methodology Process Model (Peffers et al, 2008)

Objectives

- A methodology that should:
  - be generic regarding
    - the domain of application
    - the simulator used
    - the representation of information
  - deliver increased reliability of simulation results
  - simplify and reinforce system model evaluation
    - avoid incremental simulation code development
    - minimize tools used by the user
  - enable requirements verification in the modeling tool

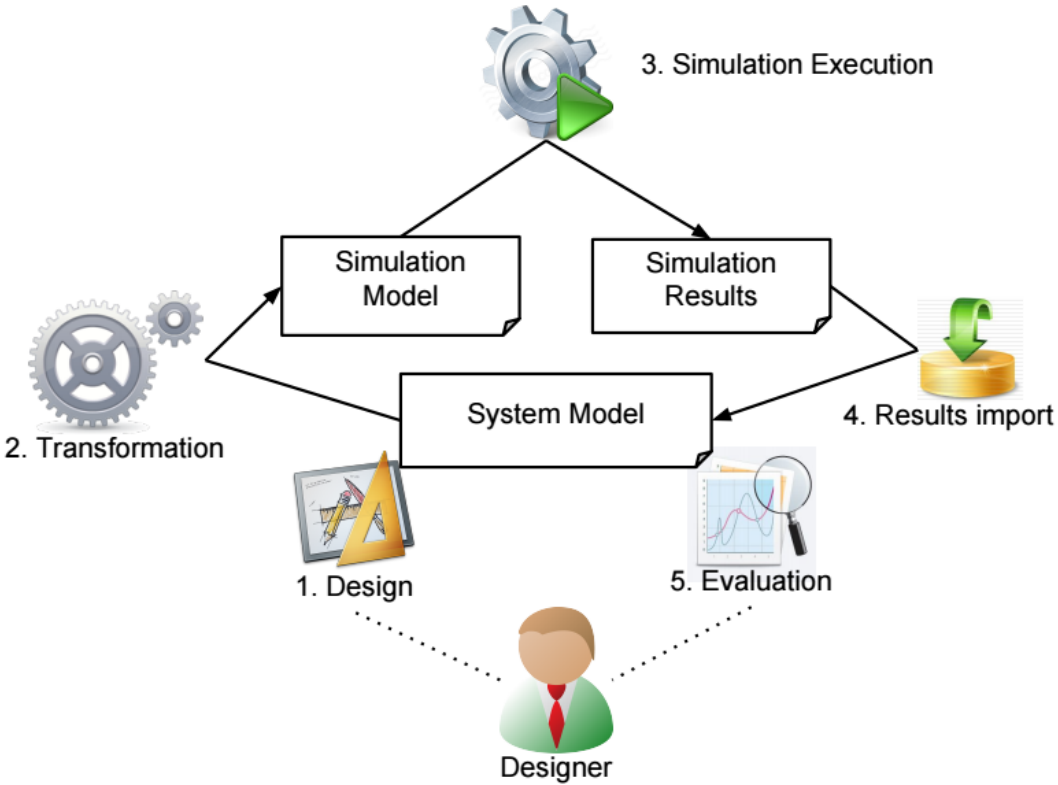
Objectives

- Specifications for frameworks that should:
  - support full automation in the generation of simulation code
  - have minimum dependence on proprietary solutions
  - support incorporation of simulation results in the system model

Proposed Methodology

1. **Design:** System model with adequate description of structure and behavior
2. **Transformation:** The system model is automatically transformed into an executable simulation model
3. **Simulation Execution:** Simulation is executed, producing results
4. **Results Import:** Simulation results are imported into predefined parts of the system model.
5. **Evaluation:** Automated requirements verification or evaluation by the designer

Proposed Methodology



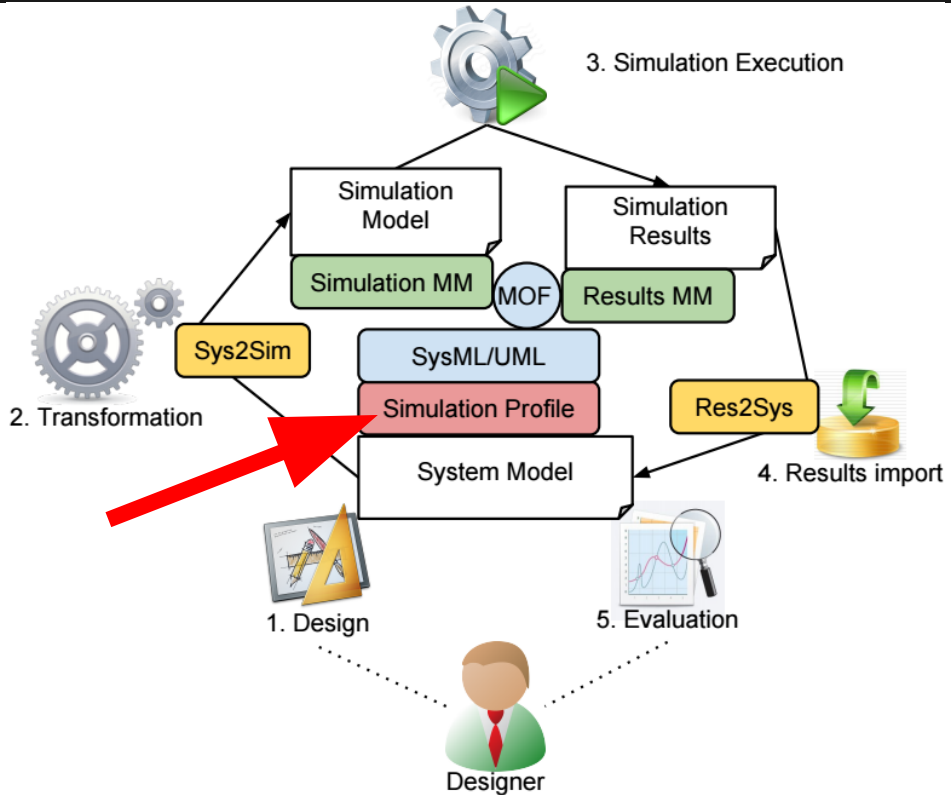
Proposed Methodology

- Simplicity:
  - The designer only:
    - defines the system model
    - is aware of simulation results
- System-Simulation model correspondence:
  - The executable simulation model is generated automatically
  - User errors are avoided and transformation errors can be identified

Framework Specifications

- Requirements & Principles:
  - Mechanisms that ensure that information required for simulation is specified correctly
  - High-level representations for simulation approaches
  - High-level transformations of system models to simulation models
  - Minimum commitment to proprietary solutions

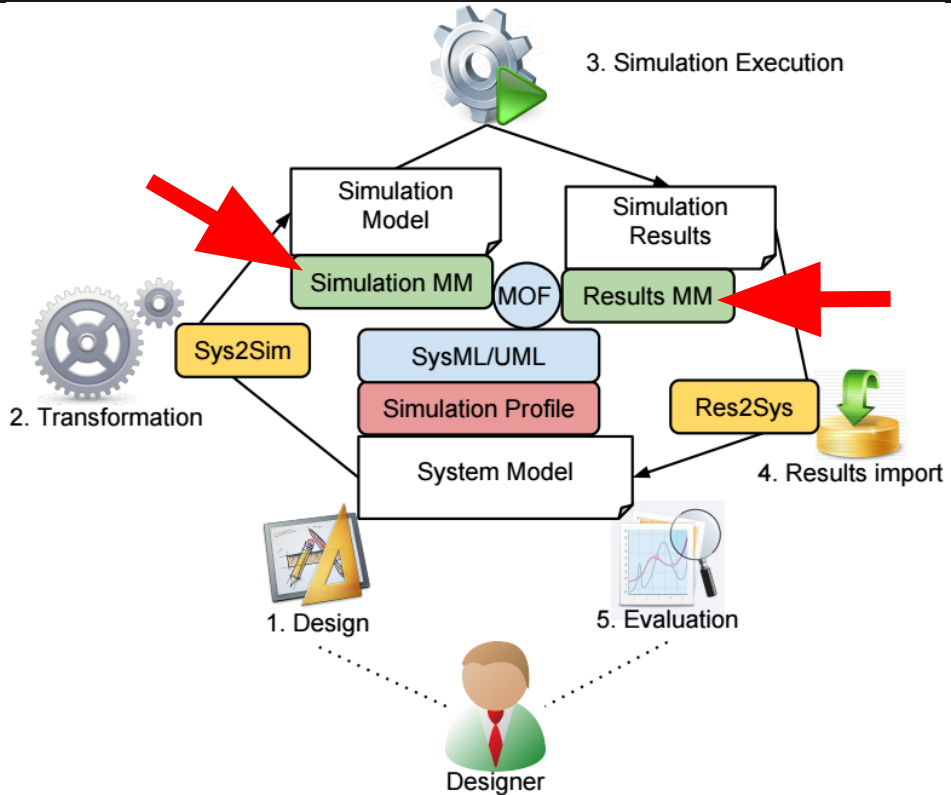
Framework Specifications



Simulation Profile

- The simulation profile extends SysML with:
  - stereotypes, to enable annotation of system model elements
  - tagged values, to provide placeholders for:
    - simulation-related information
    - simulation results
  - constraints, to check that a valid, executable simulation model can be created

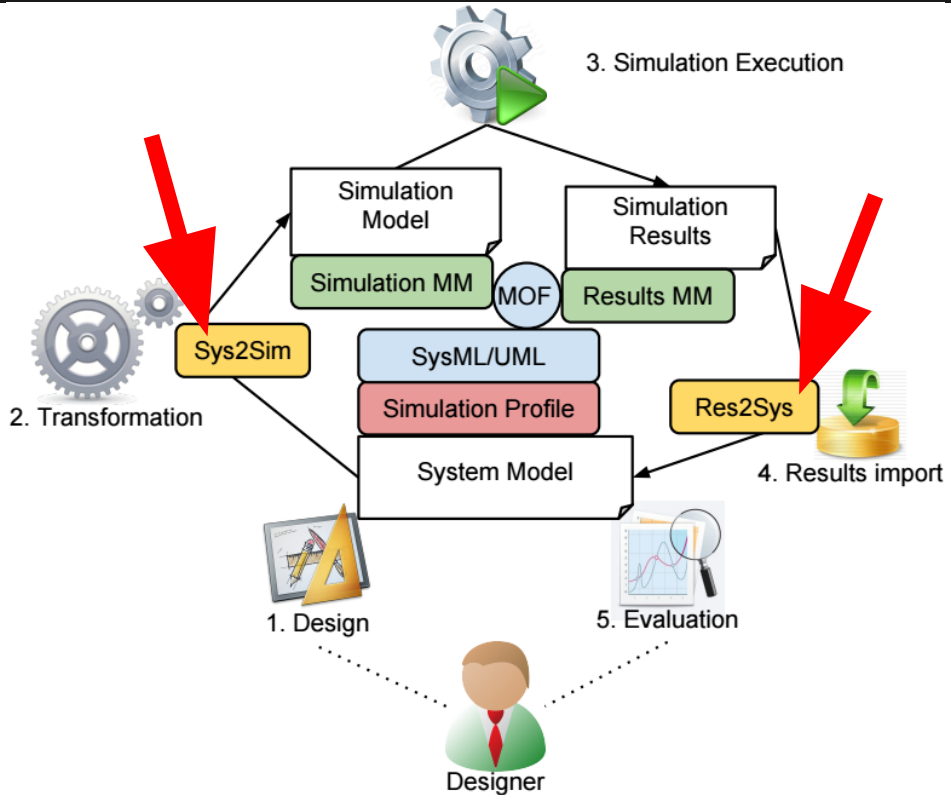
Framework Specifications



Meta-models

- The simulation meta-model:
  - is defined in terms of a solid meta-meta-modeling infrastructure (MOF), just like SysML
  - provides a simulator-independent model definition language that can be used in different simulators of a simulation approach
- The results meta-model provides the means to define simulation results

Framework Specifications



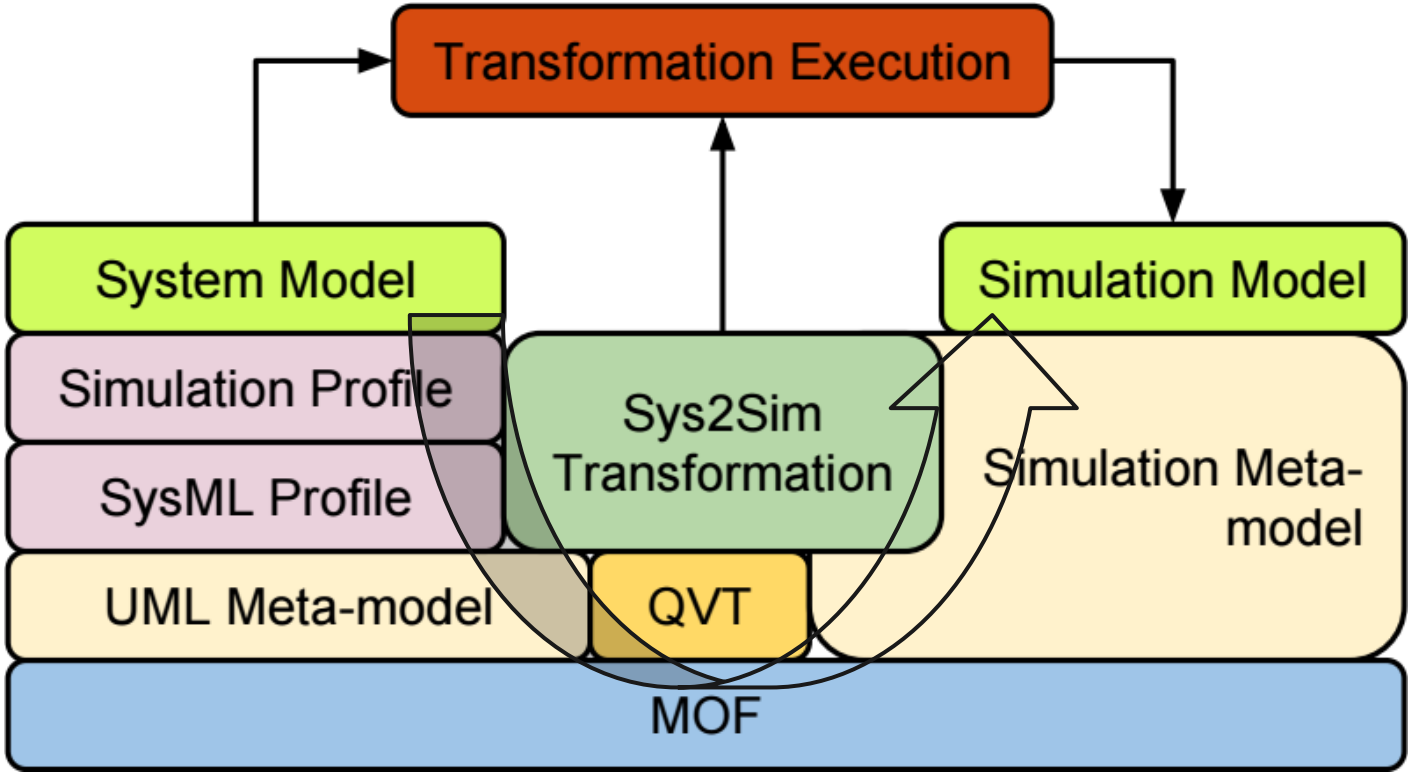
Transformations

- The system to simulation transformation:
  - locates simulation-related information
  - combines it with specific parts of the rest of the system model (e.g. model structure, interconnections)
  - generates the simulation model according to the simulation meta-model
  - is defined in terms of a transformation language that is aware of the meta-modeling infrastructure (MOF): Query/View/Transformation (QVT)

Transformations

- The results to system transformation:
  - imports simulation results in the system model
  - is defined in terms of a transformation language that is aware of the meta-modeling infrastructure (MOF)

QVT Transformation



QVT provides

- High-level transformation concepts, equivalent to system & simulation meta-models (MOF)
- Clear, distinct representation of:
  - Source model elements
  - Corresponding target model elements
  - Preconditions & postconditions
- Syntactic correctness of simulation models
- Preconditions to utilize transformation evaluation techniques (standard transformation language)

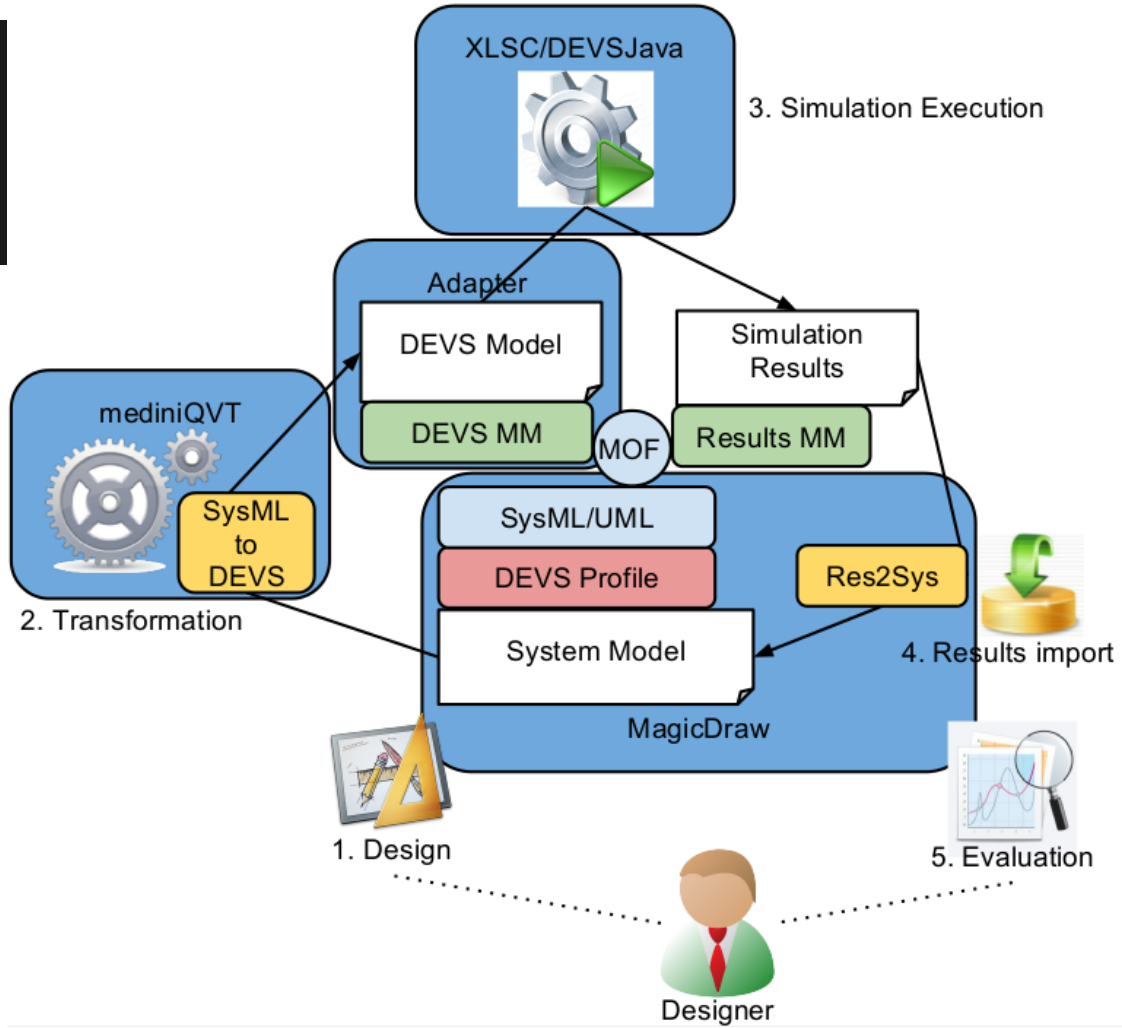
Application Guidelines

- Construct a simulation-specific profile to enrich system models with simulation capabilities
- Select/define a simulation meta-model
- Compare system and simulation model elements and create a transformation
- Adapt a simulator to support simulation model execution, if not available

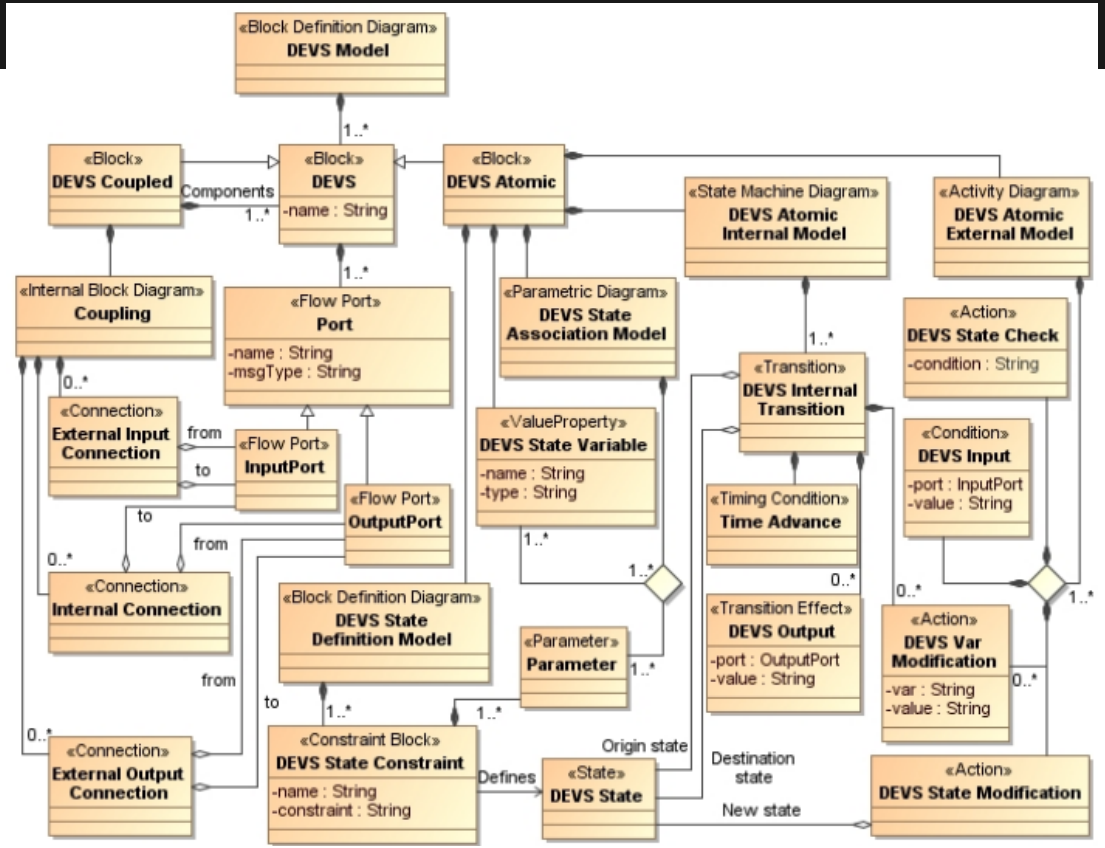
A Framework for DEVS

- A framework for simulating SysML models with DEVS has been implemented:
  - SysML profile for DEVS
  - DEVS meta-model
  - SysML to DEVS QVT-R transformation
  - Adapter to enable DEVS models execution in the DEVSJava/XLSC simulator

DEVSys Framework

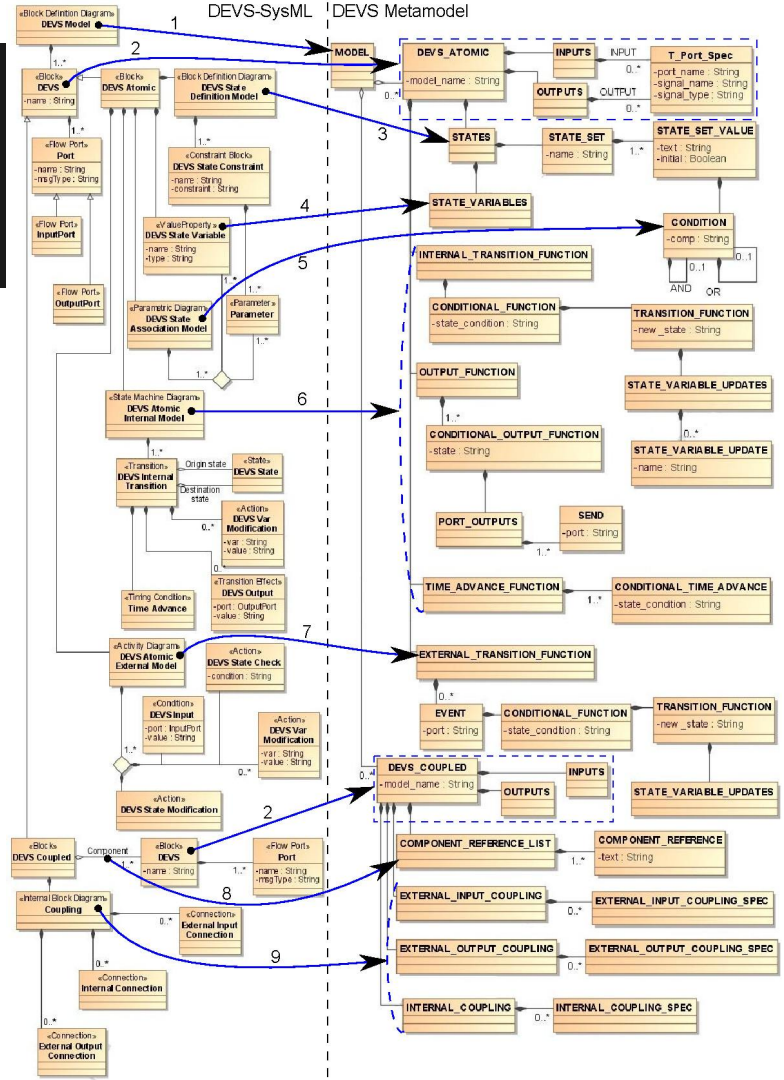


SysML Profile for DEVS



SysML to DEVS

- Correspondences are identified
- They are defined as QVT relations



Evaluation

- DEVSys has been tested in:
  - Small-scale models (<15 elements), where system behavior was defined in the system model
  - Medium-scale Enterprise Information Systems models (<500 elements), where DEVS Java simulation library components were used
- Further testing on medium and large scale models

Evaluation

- Analyze:
  - correspondence between the profile, the simulation meta-model and the transformation
  - properties of the transformation (correctness, completeness, termination)
- Study performance of transformation depending on:
  - number of system model nodes
  - number of associations
  - number of relations

Contribution

- A generic methodology for automated simulation of SysML models has been proposed
  - Systems engineers focus on the system model
    - Design
    - Simulation results
  - Consistency and correctness of simulation models
  - Enables automated requirements verification within the system model

Contribution

- Frameworks for the realization of the methodology have been specified
  - Based on and reinforcing interoperability
    - Simulation meta-models
  - Standards-based:
    - Support in diverse environments
    - Techniques for validation of combination of meta-models, profiles and transformations

Contribution

- The DEVSys framework has been implemented & tested for DEVS simulators
  - Feasibility
  - First evaluation results

Further & Future Work

- Extended evaluation in different domains (CPS) and large scale models
- Implement frameworks for other simulators
- Challenges in adopting an IoT perspective
- Compare implementations and derive common attributes
- Publish quantified evaluation results

Comments & Discussion

Publications: Conferences

- IEEE 9th International System of Systems Engineering Conference (IEEE SoSE 2014), Adelaide, Australia, 9-13 June 2014, **“Integrating Simulation Capabilities into SysML for Enterprise Information System Design”**, A. Tsadimas, G.-D. Kapos, V. Dalakas, M. Nikolaidou, D. Anagnostopoulos
- 8th Annual IEEE International Systems Conference (IEEE SysCon 2014), Ottawa, Canada, 31 March-3 April 2014, **“Model-based System Engineering using SysML: Deriving Executable Simulation Models with QVT” (Best Student Paper Award - Honorable Mention)**, G.-D. Kapos, V. Dalakas, A. Tsadimas, M. Nikolaidou, D. Anagnostopoulos
- 7th International Conference on Systems of Systems Engineering (IEEE SOSE 2012), Genova, Italy, 16-19 July 2012, **“Basic Guidelines for Simulating SysML Models: An Experience Report”**, M. Nikolaidou, G.-D. Kapos, V. Dalakas, D. Anagnostopoulos
- Third International Conference on Software Engineering Advances (ICSEA 2008), Sliema, Malta, 26-31 October 2008, **"A SysML Profile for Classical DEVS Simulators"**, M. Nikolaidou, V. Dalakas, L. Mitsi, G.-D. Kapos, D. Anagnostopoulos

Journal & Book Chapter

- SIMULATION: Transactions of The Society for Modeling and Simulation International, SAGE Publications, Volume 90, Issue 6, June 2014, pages 717-744, **“An Integrated Framework for Automated Simulation of SysML Models using DEVS”**, G.-D. Kapos, V. Dalakas, M. Nikolaidou, D. Anagnostopoulos
- Casas, Pau Fonseca i. "Formal Languages for Computer Simulation: Transdisciplinary Models and Applications." IGI Global, 2014. 1-458. Web. 1 Jul. 2013, Chapter **“10. An Integrated Framework to Simulate SysML Models Using DEVS Simulators”**, G.-D. Kapos, V. Dalakas, M. Nikolaidou, D. Anagnostopoulos, (pages 305-332)