Model management for Automotive Software

Marsha Chechik, Steve Easterbrook, Michalis Famelis, Rick Salay, Qiyu Zhu (University of Toronto), Robert Baillargeon (Panasonic), Sighe Wang (General Motors)

Automotive Software and Model Driven Development

- Importance of software rapidly increasing in the automotive industry. But Automotive Software faces unique issues.
- Fragile balance between demands for high quality and safety and issues arising from rapid, industrialized mass production.
- MDD is an approach that enables **high-level design** and **code generation** for rapidly changing embedded software systems in cars, while allowing for more formal approaches to **evaluating and ensuring software quality**.
- General Motors has used the approach with particular characteristics such as highly stylized models and custom code generators.

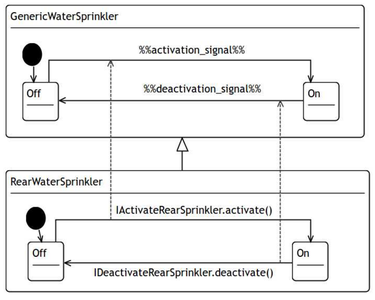
Our Goal

- We aim to create support for MDD tasks such as:
  - **Model building** and development
  - Validation and **consistency checking**
  - Traceability and **model evolution**
- Our approach:
  - Make the relations between models **first class items**.
  - To **formalize** and check them automatically.

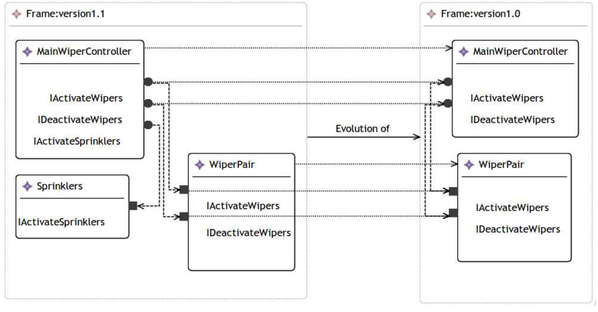
Relationships Between Models

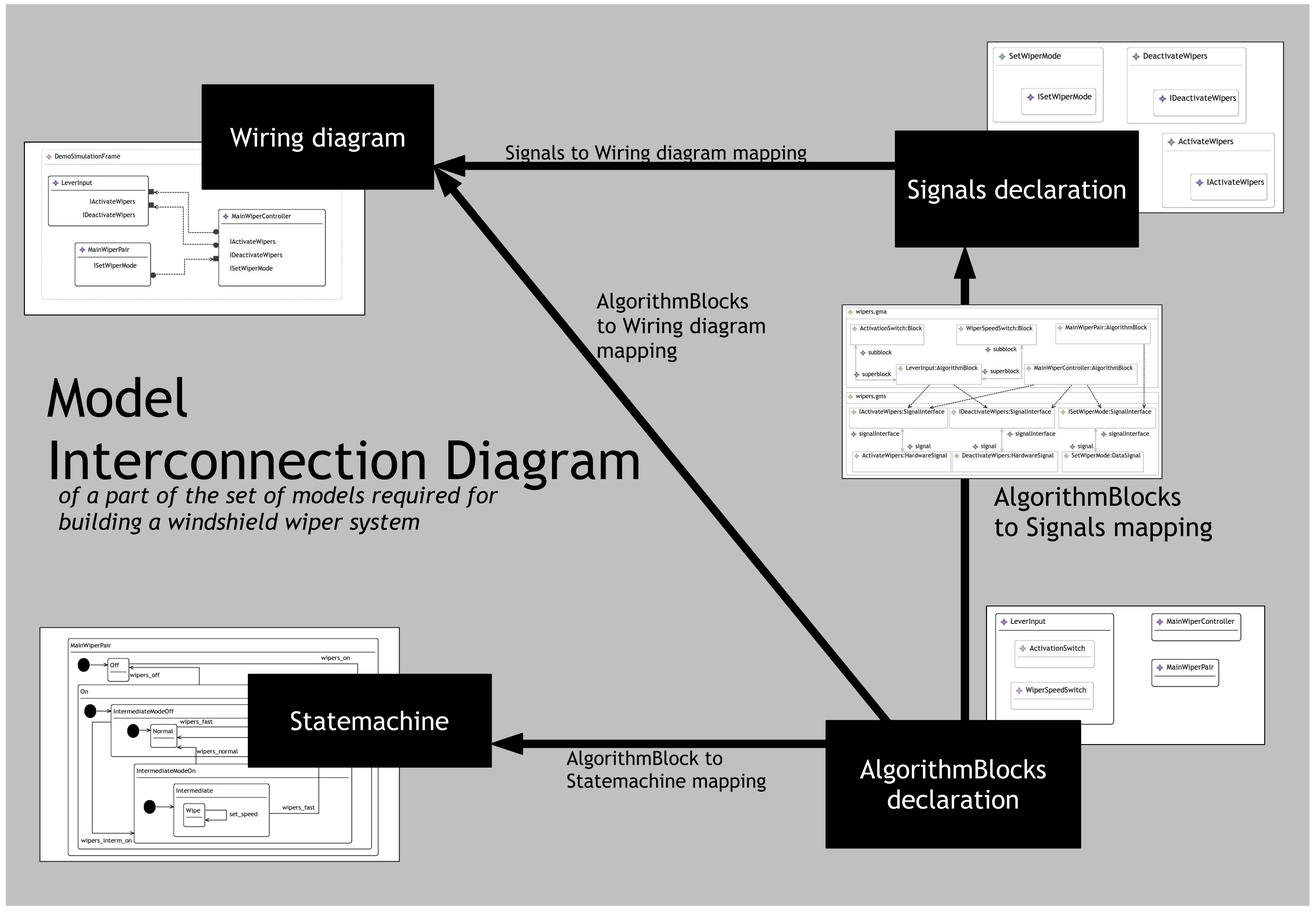
- We are exploring three types of relations in particular:
  - **Inheritance** relationship
  - Representing model **evolution**
  - Capturing information about **usage of calibration data**
- We aim to provide semantic definitions, as well as create **OCL constraints** to represent them in our framework.

Inheritance Relationship

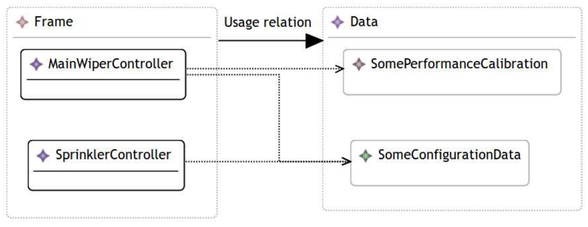


Evolution Relationship





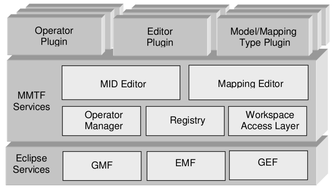
Data Usage Relationship



Model Management Tool Framework

- Eclipse-based tool framework for software model management.
  - Based on the idea of **Model Interconnection Diagrams**.
  - Support for different **types of models** and **relationships** between them.
  - Support for **consistency checking** and **operations** on Model Interconnection Diagrams.

MMTF Implementation Overview



- Support for types of models with their own GMF editors.
- OCL constrained sub-types (light types) of models.
- Relations as models using concrete syntax in Ecore.
- Support for validation of models.

Status and Future Work

- Currently formalizing the identified relations and implementing support for validating them in MMTF for couples of models that are so related.
- In the future we aim to:
  - Formalize the relations using the **QVT** framework.
  - Identify and formalize **other interesting relations**...
  - ...with additional focus on reasoning for **Software Product Lines** in the context of Automotive Software.

(The poster is available for download at: `http://www.cs.toronto.edu/~famelis/cascon09.pdf`)