Applying Data-flow Analysis to Models
A Novel Approach for Model Analysis

Meta models allow to capture the structure of an application domain in a formal and highly expressive way. However, while the use of meta models for the definition of modeling languages is a common and well-understood activity, extracting information about behavioral properties as well as the validation of static semantics is still a challenge. We present a novel approach for model analysis that addresses these issues by applying dataflow analysis to the modeling domain.

- **Motivation**
  - Meta modeling is a well established method to formally describe the structure of an application domain, facilitated by standards like the widely supported Unified Modeling Language (UML).
  - The importance of modeling raises the question of how to validate the correctness, an issue that is currently not solved satisfactorily:
    - Lack of a powerful and generic validation mechanism for models (particularly their static semantics)
    - OMG’s Object Constraint Language (OCL) does not allow to derive context-sensitive information
    - Moreover, it is not possible to perform an abstract interpretation / simulation of a model’s dynamic properties

- **Approach**
  - Data-flow analysis (DFA) is a well-understood method used in the field of compiler construction to perform an abstract interpretation of programs in order to derive static optimizations.
  - **Introduce the notion of data-flow analysis for (meta) models**
    1. Define data-flow equations on meta models
    2. Instantiate the definition for a given model
    3. Evaluate the instantiated DFA
    - Provide a meta model for DFA definitions
    - Provide an evaluation algorithm
    - Provide tooling support

- **DFA Definition and Instantiation**
  - **Definition**
    - Data-flow definitions (called Attributions) conform to AttrMM meta model
    - Specify data-flow equations using extended OCL (which allows to access data-flow values)
    - Concepts similar to Attribute Grammars (AG)
      - AttrSemanticRule: Data-flow equation for calculating an instance value
      - AttrDefinition: Data Type and Initialization Rule
      - AttrOccurrence: Occurrence of an AttrDefinition connecting a Semantic Rule to a Meta Class
  - **Instantiation**
    - Also based on a meta model (AttrM)
    - Connects attribution, model elements and results

- **DFA Evaluation**
  - Traditional DFA evaluation approach not applicable because
    - Models have no inherent flow direction
    - Models have no inherent flow direction
  - Execute rules recursively and build dependency graph
  - **DFA Attribute and Semantic Rule**
    - Cyclic input dependencies are replaced by virtual nodes (red)
    - Trigger bottom-up reevaluation for cyclic dependencies
    - Repeat reevaluation until fix-point has been reached

- **Use Case**
  - Use data-flow analysis as a generic “programming language” to implement algorithms for business process decomposition, validation and simulation, for example:
    - Determine reachability and liveness of business actions
    - Detect strongly connected components (SCC)
    - Use token-flow analysis to create hierarchical component tree
    - Apply heuristics to components to validate correctness
    - Calculate minimal/maximal paths to estimate runtime behavior

- **Implementation**

References

© Christian Saad and Bernhard Bauer, Programming Distributed Systems Lab, University of Augsburg, 2010