Systemic approach towards model definition

Model transformation semantics
Motivation

- **2 ways of model representation:**
  - Action centric (UML activity diagrams, BPMN ..)
  - Data centric (UML class diagrams)

- **Our observation:**
  - Data structure and behavior are loosely-coupled
  - Model of data structure is not related to the model of behavior
  - Model of behavior implicitly related with the model of data structure via constraints
  - Semantic of model transformation is not clear
  - Example: Within one model of the workflow information about objects is hidden inside the actions...

- **Systemic approach**
  - System thinking puts emphasis to the integration of 2 aspects within one model
  - Definition of behavior as a state change
  - Re-definition of property explicitly affects behavior
  - Re-definition of behavior explicitly affects property
Solution

- Part I
  - Relation semantics
    - Behavior-Property relations
    - Property-Property relations
  - Semantics of Model transformation
    - Refinement
    - Composition

- Part II
  - Types vs. instances:
    - Instance information embedded to relations
  - AsmL operational semantics
    - Automated model translation to AsmL
    - Simulation

- On-going:
  - Automated model verification using asmlt
Cinema X develops a web site to provide clients with new services:

A Booking tickets service enables tickets reservation via internet.

To book tickets a client has to log in on the web site. If logged in, the client can add reservations for a movie of his/her choice to a virtual cart.

The movie can be chosen from an agenda - the movie list.

The web site has to control a number of places (seats) available for every movie.

Booking tickets finishes when the client commits and logs out.
“Rectangular” vs. “Circular” community

- **Data-centric view**
  - Agenda
    - nMovies: Integer
  - ClientAccountList
    - nClients: Integer
  - Agenda
    - nMovies = 10
  - ClientAccountList
    - nClients = 54

- **Action-centric view**
  - BookingTickets
    - WebSite
      - BookingTickets
        - Customer
Systemic approach 1/3
Integration of structural and behavioral aspects within a model
1. Environment and Boundary
2. Context
3. Behavior-Property* relations
4. Stateless and Stateful properties

*Here we use the term “property” to reason about information objects presented in the system rather than a commonly used term “structure”. The latter can be used to define both “data structure” and “structure of behavior” and need to be specified.

Irina Rychkova, EPFL-IC-LAMS
Systemic approach 3/3

- **Boundary**: A border between the system (a “Hollon”) and its environment.

- **Environment** (of an object): The part of the model which is not the part of that object. [RM-ODP]

- **Context** (modeling context): A scope of interests for a given model. Model elements (properties, behaviors, relations) can be specified as relevant or irrelevant to a given context. LifeCycle context is a global context for any system. Context can be defined by the hollon and its behavior of interest.
**Property**

**Stateful property**: a property who’s instance state can be directly obtained and interpreted as a value of one of the standard operational types*.

**Stateless property**: a property who’s instance state is not defined and can be expressed only indirectly, via states of related properties (attributes).

* Standard operational type: here, Integer, String, Char, Long, Double, Boolean.
Part I:

- Semantic definition for relations
Relating Behavior to Data Structure

- **UML: OCL constraints**
- **SEAM:**
  - Property-Property relations;
  - Behavior-Property relations;
  - Action-Action relations
Property-Property relations

- **Exhibit relation**: relation between a stateless property (the owner) seen as atomic and its attribute (in general also can be a stateless property) that characterizes the property at a given abstraction level.

- **Composition relation**
- **Association relation**:

```
<cardinality>|<instance_name>
```
Behavior-property relation: 1. A model element that expresses a semantic relation between action and property; specifies behavior effect in terms of state change. 2. The way to define a) a precondition, b) an emergent property or c) a postcondition for an action.
Behavior- Property relations: Precondition

- **System preconditions** (PreS) define predicates in terms of system variables obtained as a result of a previous system activity.

- **Environment preconditions** (PreEnv) define predicates in terms of parameters, provided by the environment via events.
Behavior- Property relations: Emergent property

A predicate that specifies the action termination. (true-false, ok-error, etc). Based on the Emergent property, action effect can be chosen. In general case an emergent property might have more than 2 values.
Behavior - Property relations: Postcondition

- **System postcondition** (PostS) specifies the action effect hidden from the environment.

- **Environment postcondition** (PostEnv) specifies the action effect and the way to make it visible and/or accessible for the environment (events declaration etc).
Action-Action relations

- UML
- BPMN
Model transformation

- Semantics for model transformations:
  - Refinement
  - Composition
Refinement 1/3

Class redefinition:

ClientAccountList

How it affects the definition of BookingTicket action?

OCL expressions: ...
ClientAccountList redefined → BookingTickets must:

1. **choose the ONE client out of the list**
2. **modify the reservation list of this client**
3. **Accept corresponding parameters from environment**
Refinement 3/3

BookingTickets redefined →

1. Relations to properties must be redistributed between component actions
2. Emergent property required
3. Communication with environment needs revision
Semantically safe transformations

- Binding expressions:

\[ \text{cList.nClients} = \text{cList'}.\text{size()} \]
Benefits

- Semantically rich models
- Explicit Behavior-changes-state notation
- Simulation possible
Part II

- Application
Types vs. Instances

- Instance information is embedded into the model:
  - Property-Property relation links instances (ordered sets of instances in case of * cardinality)
  - Behavior-Property relation defines an operation via instances
  - Binding using instance names

- Values:
  - For every statefull property its initial state predefined
SEAM – AsmL translator

- AsmL – abstract model specification language with environment for simulation and testing
Tool for automated code generation

SEAM model

SEAM meta-model

Transformer

Java

ASML model

ASML meta-model

Irina Rychkova, EPFL-IC-LAMS
Translation/Transformation

- **Transformation 1:**
  - From Graphical SEAM (XML) to SEAM abstract syntax tree (AST).

- **Specification of the Model_Of_Interest**
  - Hollon
  - Behavior of Interest
  - Level of Details

- **Transformation 2:**
  - From SEAM AST to AsmL AST

- Printing out AsmL AST to the file, executable by AsmL tool.
Benefits

- Verification of refinement “on the fly”
- Correct by construction models
Future work

- Refinement verification using Asmlt.
- Expressions in SEAM likely to be written in some platform-neutral language (currently written in AsmL) with the possibility to parse and translate to many platforms (asml, ocl, java, scala..) for further verification and/or simulation.
- Semantic bindings for model refinement are detected by the translation tool however cannot be defined automatically.
  - Automated analysis of Expression refinement together with Property ans Behavior refinement is required.
  - QVT?
Presentation structure

- **Theoretical problem:**
  - There are 2 ways to look at the world – rectangular and circular.
  - In modeling 2 semantics exist:
    - Actions within activity and processes
    - Objects within data structure
  - Structure and Behavior are loosely-coupled.

- **Practical problem:**
  - Within one model of the workflow information about objects is hidden inside the actions…

- **Theoretical Solution:**
  - System thinking puts emphasis to the integration of these 2 aspects within one model.
  - If both aspects are visible on the same model ->
    - Definition of behavior as a state change
    - Re-definition of property explicitly affects behavior
    - Re-definition of behavior explicitly affects property

- **Practical solution:**
  - Semantic relations between behavior and structure.

- **Impact:**
  - Semantic of model transformation
  - Verification