Formal Semantics for Property-Property Relations in SEAM Visual Language:
Towards Simulation and Analysis of Visual Specifications

Irina Rychkova and Alain Wegmann,
School of Computer and Communication Science,
Lausanne, Switzerland
Outline

1. Context
   - Visual Modeling in Enterprise Architecture - SEAM
   - Benefits of formalization
   - Problem

2. Knowledge base
   - Theories used for model formalization

3. Research Result
   - Formalization of property-property relations in VM
   - Criteria of model consistency
   - Application
1. **Context**

SEAM method for Enterprise Architecture
Enterprise Model

- 4 levels:
  - Value networks (group of companies)
  - Companies
  - IT System
  - Web services

- 2 analysis:
  - As-is
  - To-be

Method: SEAM © LAMS - EPFL
Enterprise Model

- 4 levels:
  - Value networks (group of companies)
  - Companies
  - IT System
  - Web services

- 2 analysis:
  - As-is
  - To-be
SEAM: Overview

- One diagram type
- Hierarchical structure of models
- Systems (called working objects) represented
  - as a whole: properties (data structure) + behavior
  - as a composite: system construction
SEAM Metamodel

System Data Structure

Property

Property-Property Relation

Action

Action-Action Relation

Activity

System Behavior
Compound properties are defined by a set of component properties and/or references to properties using property-property (PP) relations.
Visual vs. Formal semantics

Visual Specifications:

- Suitable for documentation and discussion between people with different backgrounds
- Easy to read and understand by a user
- Require a little training from a developer
- Support a high level of abstraction
- Lack of precision

Formal semantics is needed
Formalization Benefits

- Model Simulation
- Model Comparison
- (Automated) Model Development

Visual Model → Formal Semantics

Formal Semantics →
- Mapping Rules
- Invariants/Assertions
  - Correctness/Consistency
  - Validation
- Executable Code
- Tests
- Proof Obligations
- Simulation
- Comparison

(Automated) Model Development
Problem:

- Consistency checking
Solution:

- Formalization of PP-relations
2. Knowledge Base

Theories to support formalization
Theories

- Higher-Order logic
- Refinement Calculus
  [Back, R.-J., von Wright, J. 1998]
- Relation Partition Algebra (RPA)
  [Feijs, L.M.G., van Ommering, R.C, 1999]
- Theory of Multi-Relations
  [Feijs, L.M.G., Krikhaar R.L.,1998]
Theories

- Higher-Order logic
- Ref. Calc.
- RPA
- Th. of MR

**Formal Semantics:**
- Extension of Relation partition algebra + theory of multi-relations

**System Data Structure**

**Activity**

**System Behavior**
Theories

- Higher-Order logic
- Ref. Calc.
- RPA
- Th. of MR

Formal Semantics: Extension of Relation partition algebra + theory of multi-relations

MSWEIS 2007
3. Research Result

Towards model correctness and consistency
Property-Property relations

RPA formalization
RPA

- Relation Partition Algebra (RPA) defines 'part-of' and 'use' relations as a special type of binary relations. These terms can be used to represent PP-relations in SEAM.

- The modularization of the data types gives rise to the **part-of relations**.

- Mechanism, when one property references (uses) another, can be modeled by **use relations**.
SEAM multi-relations sm

- Definition of Part-of and use relations with multiplicities: \[ sm_{sup}(P,Q), \ sm_{inf}(P,Q) \]

- Properties

- Mathematical reasoning:
  - Relation composition \( sm(P,Q) \circ sm(Q,R) \)
  - Exponential \( sm_{sup/inf}(P,Q)^n \)
  - Identity relation \( I = sm(P,P) \)
  - Transitive closure \( sm^+(P,Q) \)
A part-of relation between properties P and Q specifies that:

- property P is a part of a data type, defined by property Q
- there are at least \(n_1\) and at most \(n_2\) instances of property P for each instance of Q

\[
\text{part}_{\text{inf}}(P,Q) = n_1, \quad \text{part}_{\text{sup}}(P,Q) = n_2 \iff \\
\forall y : Q \exists x_{n_1}, \ldots, x_{n_2} : P \mid n_1 \leq n \leq n_2
\]
SEAM ‘Use’ relation

- A use relation between properties P and T specifies that:
  - property P references property T;
  - there are at least $r_1$ and at most $r_2$ instances of property T used (or referenced) by each instance of property P.

\[
\text{use}_{\inf}(P,T) = r_1, \text{ use}_{\sup}(P,T) = r_2 \iff \\
\forall x: P \exists z_{r_1}, \ldots, z_{r_2} : T \mid r_1 \leq r \leq r_2
\]
Consistency Criteria
Problem:
**Part-Of relations: Consistency Criteria**

- Is functional: Property $P$ can be a part of one and only one compound property
  \[ \forall P, Q, R \in P \text{ part}^\text{sup}(P, Q) > 0 \land \text{ part}^\text{sup}(P, R) > 0 \iff Q = R \]

- Is cycle-free: there is no path that starts at $P$ and leads back to $P$
  \[ \forall P \in P \text{ part}^+(P, P) = \emptyset \]
Use relations: Consistency Criteria

- Is non-functional - can be referenced by multiple compound properties
  \[ \exists T, P, P' \in P \mid P \neq P' \bullet \text{use}^\text{sup}(P, T) > 0 \land \text{use}^\text{sup}(P', T) > 0 \]

- Can be cyclic - can be referenced by itself
  \[ \exists T \in P \mid \text{use}^+(T, T) \neq \emptyset \]
- Maximum and minimum number of instances of P in the system:
  \( \text{Inst}_{\text{max}}(PC), \text{Inst}_{\text{min}}(PC) \)

- Free-floating property: \( \text{Inst}_{\text{min}}(P) = 0 \)

- Maximum number of references to T in the system:
  \( \text{Ref}_{\text{max}}(PC) = 40 \implies \text{Inst}_{\text{max}}(PC) \geq 40 \)

- Instantiation efficiency: \( \text{Ref}_{\text{max}}(P) \geq \text{Inst}_{\text{max}}(P) \)
Consistency criteria: Summary

- System data structure defined by SEAM specification is consistent if:
  - all part-of relations in the specification are functional and cycle-free
  - instance declaration is sufficient:
    \[ \forall P \in P \ \text{Ref}_{\text{max}}(P) \leq \text{Inst}_{\text{max}}(P) \]
  - no reference on a free-floating property exists
    \[ \forall P \in P \ | \ \text{Inst}_{\text{max}}(P) = 0 \Rightarrow \text{use}(P, P) = \emptyset \ \forall P_i \in P \]
Application

Refinement Propagation
The refinement propagation technique for SEAM visual specifications explores the possible conflicts between model elements, caused by modifications (model refinement) and applies specific rules to enforce the model correctness.

Automated model refinement is the main practical benefit expected from the proposed formal semantics.
Refinement Propagation

- The refinement propagation technique for SEAM visual specifications explores the possible conflicts between model elements, caused by modifications (model refinement) and applies specific rules to enforce the model correctness.

- Automated model refinement is the main practical benefit expected from the proposed formal semantics.
The refinement propagation technique for SEAM visual specifications explores the possible conflicts between model elements, caused by modifications (model refinement) and applies specific rules to enforce the model correctness.

Automated model refinement is the main practical benefit expected from the proposed formal semantics.
Summary

- SEAM modeling language in context of the EA
- Formalization of PP-relations in SEAM
- PP-relations + RPA = SEAM ‘part-of’ and ‘use’ relations
- Consistency criteria
- Automated model refinement
Formal semantics for Visual Notation

- Combination of visual and formal methods:
  - OPM
  - DEMO
  - BPMN, BPEL (see Intalio, IBM Websphere, etc)
  - UML + OCL
  ...

Irina Rychkova, Alain Wegmann
MSVEIS 2007