Abstract Syntax Can Make the Definition of Modelica Less Abstract

1st International Workshop on Equation-Based Object-Oriented Languages and Tools

Berlin
July 30, 2007

David Broman and Peter Fritzson
Linköping University, Sweden
{davbr,petfr}@ida.liu.se
Outline of the Talk

Part I - Why are specification improvements needed?

- Specification
- Goals and Problems

Part II - What to specify?

- Transformation Aspect
- Rejection Aspect

Part III - How to specify?

- Different Approaches
- Previous Attempts
- Abstract Syntax Approach

Part I
Why are specification improvements needed?

Part II
What to specify?

Part III
How to specify?
Part I

Why are specification improvements needed?
Problem: Interpreting the specification

Problem
- The Modelica Specification is open for interpretation
- May result in incorrect and incompatible tools

Part I
Why are specification improvements needed?

Part II
What to specify?

Part III
How to specify?
Specification goals

**Unambiguous**
"Can only be interpreted in exactly one way"

**Understandable**
"Easy to grasp with moderate computer science knowledge"

**Expressive**
"State the syntax and semantics in a compact manner"
An evolutionary "middle-way" strategy

Revolutionary Strategy
Very large specification.

Evolutionary Strategy
Abstract Syntax Approach

Informal approach
e.g. natural language text

Formal approach
e.g. operational semantics

\[
\Gamma \vdash e_1 : \text{bool} \quad \Gamma \vdash e_2 : T \quad \Gamma \vdash e_3 : T \\
\Gamma \vdash \text{if } e_1 \text{ then } e_2 \text{ else } e_3 : T
\]

Part I
Why are specification improvements needed?

Part II
What to specify?

Part III
How to specify?
Part II

*What* to specify?
**What should be specified?**

- **Syntax** - the structure
- **Semantics** - the meaning

1. **Transformation Aspect**

   ![Modelica Model](image1)

   Simulation Result

2. **Rejection Aspect**

   ![Modelica Model](image2)

---

**Part I**
Why are specification improvements needed?

**Part II**
What to specify?

**Part III**
How to specify?
Transformation Aspect

What is actually the result of an execution?

```
<table>
<thead>
<tr>
<th>Part I</th>
<th>Part II</th>
<th>Part III</th>
</tr>
</thead>
<tbody>
<tr>
<td>Why are specification improvements needed?</td>
<td>What to specify?</td>
<td>How to specify?</td>
</tr>
</tbody>
</table>
```

![Diagram showing the process from Modelica model to Simulation Result](image)

- **“Static” semantics / compile time**
  - Modelica model → Elaboration → Hybrid DAE
    - Informally specified in the Modelica Specification
  - Hybrid DAE → Equation Transformation & Code generation
  - Executable

- **“Dynamic” semantics / run time**
  - Executable → Simulation
  - Simulation Result
Rejection Aspect

What is actually a valid Modelica model?

Part I
Why are specification improvements needed?

Part II
What to specify?

Part III
How to specify?
Part III

How to specify?
Previous Specification Attempts

**Types in Modelica**  
(Broman, Fritzson & Furic, 2006)  
- Clarify the type concept in Modelica  
- Concerns only the rejection aspect

**Instance Creation (Elaboration)**  
(Mauss, 2005)  
- Only transformation aspect  
- Subset of the language

**Modelica Specification**  
(Modelica Association, 2005)  
- Informal semantics, natural language  
- Concrete syntax

**RML and Natural Semantics**  
(Kågedal & Fritzson, 1998)  
- 1998, large subset of specification  
- hard to get an overview of - became very large  
- now the code base for OpenModelica

**Informal approach**  
-e.g. natural language text

**Abstract Syntax Approach**

**Formal approach**  
e.g. operational semantics

\[
\Gamma \vdash e_1 \colon \text{bool} \quad \Gamma \vdash e_2 \colon T \quad \Gamma \vdash e_3 \colon T \\
\Gamma \vdash \text{if } e_1 \text{ then } e_2 \text{ else } e_3 \colon T
\]  
(t-if)

---

**Part I**  
Why are specification improvements needed?

**Part II**  
What to specify?

**Part III**  
How to specify?
Abstract Syntax as a Middle-Way Strategy

Scanning, Parsing => Abstract Syntax Tree (AST)

Part I
Why are specification improvements needed?

Part II
What to specify?

Part III
How to specify?

Elaboration (transformation and rejection aspects)
- Input: Implicitly specified using concrete syntax
- Output: Not specified
- Transformation: Informally specified using natural language
Transformation and Rejection Aspects

**Rejection Aspect**
- More restrictive than the concrete syntax grammar
- Include context sensitive information
- Still need informal rules (e.g. type rules)

**Transformation Aspect**
- Precise specification of input and output (grammar)
- Transformation still informal

---

Part I
Why are specification improvements needed?

Part II
What to specify?

Part III
How to specify?

David Broman, davbr@ida.liu.se
Example of a grammar fragment (S-AST)

Extended BNF style

States allowed local class types and component types

States allowed prefixes

Meta-variables, declarative context-sensitive information

Verbose - intended for specification, not implementation

Key points

- Extended BNF style
- States allowed local class types and component types
- States allowed prefixes
- Meta-variables, declarative context-sensitive information
- Verbose - intended for specification, not implementation

Modelica specification about connectors:
"No equations are allowed in the definition or in any of its components"

```
connector ::= Connector(
    {Extends(Cr conModification)}
    {DeclCon(modifiability outinner Cd connector)}
    {DeclRec(modifiability outinner Ra record)}
    {CompCon(conconstraint Cr cd conModification)}
    {CompRec(conconstraint Rr rd recModification)}
    {CompInt(conconstraint xd)}
    {CompReal(conconstraint flowprefix yd)}
)

access ::= Public | Protected
modifiability ::= Replaceable | Final
outinner ::= Outer | Inner | OuterInner | NotOuterInner
conconstraint ::= Input | Output | InputOutput
flowprefix ::= Flow | NonFlow
```
Concluding Summary

Specification Goals
- Unambiguous
- Understandable
- Expressive

What to specify
- Transformation aspects
- Rejection aspects

How to specify
- Current specification is informal, e.g. open for interpretation
- Evolutionary "middle-way" approach

Abstract Syntax Approach
- Specify INPUT and OUTPUT using grammars
- Complements the informal semantics

Next Step
- Needs to be accepted by Modelica Association as the fundamental approach.
- Yet another "specification" of a subset is less meaningful
- There can only be one specification...

Thanks for listening!

Part I
Why are specification improvements needed?

Part II
What to specify?

Part III
How to specify?