Consistency Analysis of UML Class and Sequence Diagrams using Attributed Graph Grammars

Aliki Tsiolakis, Hartmut Ehrig
Technische Universität Berlin

25 March 2000
Contents

• UML diagrams and consistency
• Relations between UML class and sequence diagrams and resulting consistency problems
• UML class and sequence diagrams as attributed graphs and graph grammars
• Algorithms for consistency analysis
• Conclusion and open issues
UML Diagrams and Consistency Analysis

Unified Modeling Language (UML)

System

static structure
interaction between objects
internal behavior
run-time configuration

Problem: Consistency of different views
**UML Class Diagrams**

- **Classes**
  - Attributes
  - Operations
- **Relations**
  - Associations
  - Aggregations
  - Generalizations
- **Adornments**
  - Navigability
  - Multiplicities
  - Roles
UML Sequence Diagrams

- **Objects**
  - Name
  - Role
  - Class name

- **Messages**
  - Between existing objects
  - **Creation of objects**
  - Destruction of Objects
Relations between Class and Sequence Diagrams

Existence checking

Multiplicity checking
Attributed Graphs and Graph Grammars

- Object graphs are typed over the class graph
- Algebraic attributes for data type aspects

**UML class diagram**

\[\downarrow\text{represented as}\]
- attributed graph
- attributed graphical constraint

**UML sequence diagram**

\[\downarrow\text{represented as}\]
- attributed graph grammar
- additional control condition
Representation of a Class Diagram as Graph and Graphical Constraints

A
methods = {}

relation

B
methods = {("createB")}

relation

C
methods = {("createC")}

Representation of the class diagram

Representation of a multiplicity

lower bound (n = 1)

upper bound (m = 1)
Representation of a Sequence Diagram as Graph Grammar

start graph:

\[
\begin{align*}
\text{A} & : \text{methods} = \text{Ma} \\
\text{B} & : \text{methods} = \{\text{“createB”}\} \\
\text{C} & : \text{methods} = \{\text{“createC”}\}
\end{align*}
\]

rules:

\[
\begin{align*}
\text{R1} & : \text{A methods} = \text{Ma} \\
\text{R2} & : \text{B methods} = \text{Mb}
\end{align*}
\]

control condition: (R1;R2;R2)
Consistency Analysis

Existence Checking

Multiplicity Checking

lower bound

upper bound

Total graph morphisms:
• $L \rightarrow T$
• $R \rightarrow T$

• Technique of Heckel and Wagner
• *dangling condition* required
Consistency Analysis —
Multiplicity Checking

Unsuccessful checking of an association’s lower bound
Conclusion

• Formal approach for consistency checking of UML diagrams based on attributed graphs and graph grammars

• Prototype implementation for consistency checking using AGG

• Consistency checking between further UML diagram types using other formal approaches
  – between state charts and sequence diagrams
Open Issues

• Formal extension of Heckel-Wagner algorithm for attributed typed graphs

• Extension of consistency checking
  – Constraints for consistent states in sequence diagrams
  – Pre and post conditions of sequence diagrams that are not marked as complete
  – Additional model elements for class and sequence diagrams
    • Control structures in sequence diagrams
    • Stereotypes
    • ...
Consistency Analysis of UML Class and Sequence Diagrams using Attributed Graph Grammars

Aliki Tsiolakis, Hartmut Ehrig
Technische Universität Berlin

{aliiki, ehrig}@cs.tu-berlin.de