Toolkit for Conceptual Modeling (TCM)
Specification for TUCD
(Use Case Diagram editor)

version 1.2

Frank Dehne
Faculty of Mathematics and Computer Science
Vrije Universiteit
De Boelelaan 1081a, 1081 HV Amsterdam
The Netherlands
tcm@cs.vu.nl

Henk R. van de Zandschulp
Department of Computer Science
University of Twente
P.O. Box 217, 7500 AE Enschede
The Netherlands
tcm@cs.utwente.nl

June 5, 2000

Contents

List of Figures

Chapter 1

Use Case Diagram Editor (TUCD)

1.1 Specification

1.1.1 Terminology

None.

1.1.2 Restrictions and Extensions

- **Restrictions / Not implemented**

  The following constructs are not implemented because their semantic is unclear and consequently they are difficult to use correctly.

  - **Use Cases:**
    - no extension point within use cases.
  - **Actors :**
    - no attributes
    - no events
  - **Relationships :**
    - no ≪include≫ relationships
    - no ≪extend≫ relationships

Figure 1.1: Use Case Diagram restrictions.

- **Extensions of TGD**
  - system
  - binary associations between actor and use case
  - generalization between actors
  - generalization between use cases
  - change actor type between StickMan and Class symbol

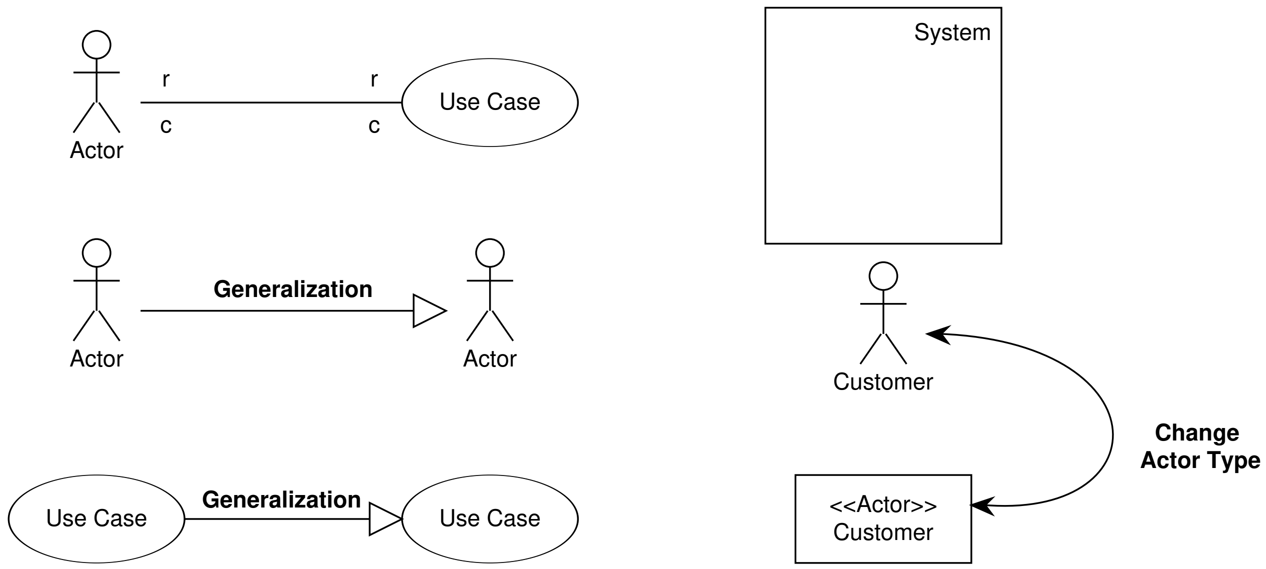

Figure 1.2: Use Case Diagram extensions.

1.1.3 Abstract Model

- **Nodes**

  - Actor
  - Use Case
  - System

  See figure 1.3.

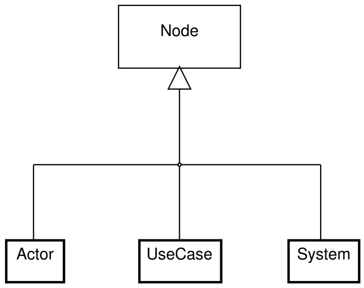

Figure 1.3: Use Case Diagram nodes.

- **Edges**
  - Binary Association, with optional rolenames and multiplicity adornments.
  - Generalization, both between Actors and Use Cases

  See figure 1.4.

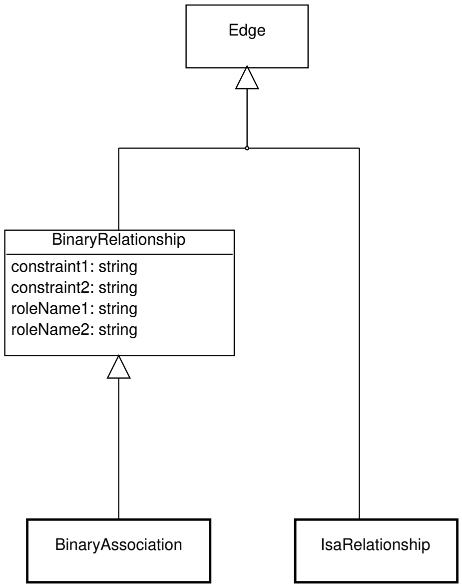

Figure 1.4: Use Case Diagram edges.

- **Consistency rules**

  See figure 1.5.

Constraint	Description
UC1	Names of actors and use cases contain at least one non−space character.
UC2	Specialization relationships should not contain cycles.
UC3	Actors and use cases should have mutually unique names.
UC4	Only one system per diagram is allowed.
UC5	No use cases allowed outside a system.
UC6	No actors allowed within a system.
UC7	Specialization relationships should be between two actors or between two use cases.
UC8	Binary association should be between actor and use case.
UC9	Binary association has no label (name).
UC10	Stereotype label <<Actor>> can not be changed

Figure 1.5: Use Case Diagram constraints.

1.1.4 Representation

- **Nodeshapes**
  See figure 1.6.

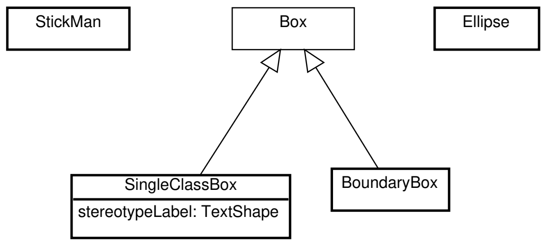

Figure 1.6: Use Case Diagram nodeshapes.

- **Lines**
  See figure 1.7.

1.1.5 Additional functions

- **Change Actor type** between StickMan and ClassBox representation of Actor

1.1.6 File format

No changes.

1.1.7 User Interface

- **Node buttons**

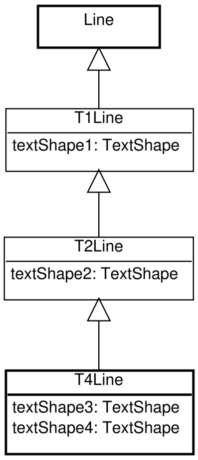

Figure 1.7: Use Case Diagram lines.

- Actor (StickMan)
- Actor (ClassBox)
- Use Case (Ellipse)
- System (Boundary Box)

- **Edge buttons**
  - Binary Association
  - Generalization

1.2 Design

1.2.1 C++ class structure changes

None.

1.2.2 C++ class diagram

See figure 1.8.

1.2.3 Consistency Checks

See figure 1.9.

1.2.4 User Interface

None.

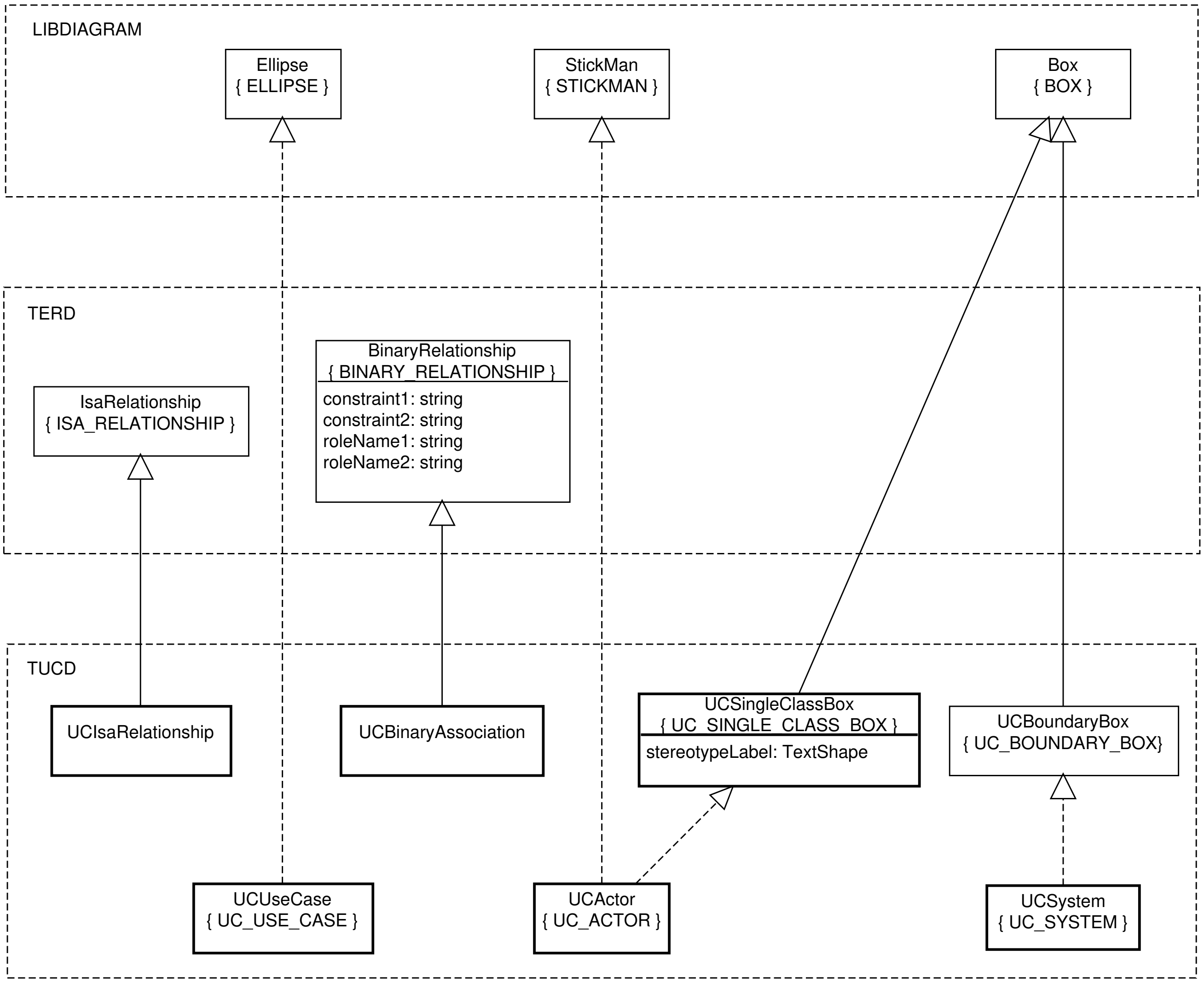

Figure 1.8: Use Case Diagram Classes

Constraint	Check
UC1	Soft
UC2	Immediately
UC3	Immediately
UC4	Immediately
UC5	Immediately/Soft
UC6	Immediately/Soft
UC7	Immediately
UC8	Immediately
UC9	Immediately
UC10	Immediately

Figure 1.9: Use Case Diagram constraints checks.

1.2.5 Files

- .../bin/tucd
- .../src/bitmaps/StickMan.xbm
- .../src/dg/stickman.[ch]