A UML Semantics FAQ: The View from Bremen

Martin Gogolla, Oliver Radfelder, Mark Richters
University of Bremen, FB3, Computer Science Department

May 17, 1999

Abstract

This note spells out questions the authors found while studying the material defining the Unified Modeling Language UML. After formulating a preliminary question in Sec. 1, some detailed questions with answers and some detailed questions without answers are given in Sec. 2 and 3, respectively. Sections 4 and 5 just state questions with and without answers but we do not go into details explaining the importance of the questions. Our answers to the 'questions with answers' can be found in our recent papers [GR98c, GR98b, GP98, GR98a, Gog98, RG98]. In Sec. 6, a short hint to relevant literature ends this note.

Motto

Now when the queen of Sheba heard about the fame of Solomon concerning the name of the LORD, she came to test him with difficult questions. *New American Standard Bible, Book 1 Kings 10:1,* `http://bible.gospelcom.net/bible`*, 1999.*

1 A Preliminary Question

QUESTION: What are (currently and in the future) the reference documents for UML, especially for a UML Semantics FAQ?

Without doubt one has to take into account the 'UML Notation Guide' and the 'UML Semantics'. But which version? Of course the latest stable one. But: Version 1.1 is stable however dated September 1997; following the OMG web pages, Version 1.3 is still under development however the Version 1.3a1, January 1999, delivered on CD with the book 'UML Refererence Manual' by Rumbaugh et al. does not contain an explicit note stating that that version is still developing.

Does one in addition take into account the books 'UML User Guide' by Booch et al. and 'UML Refererence Manual' by Rumbaugh et al.? How does one deal with inconsistencies between these four documents? Which document takes priority? One example for such an inconsistency is:

Signals ... have no operations. Notation Guide, page 3-127.

A signal ... may have operations. Reference Manual, page 428.

2 Questions with Answers

2.1 QUESTION 1: UML Semantics expressed within UML

QUESTION: Is it possible to express the semantics of UML within UML?

Yes, this is at least partly possible. The general approach is to structure the UML semantics problem into four steps mainly by introducing an intermediate layer for 'UML heart concepts':

1. Identify as many as possible equivalence rules for UML concepts. Such rules should state something like: the UML expressions α and β are equivalent (perhaps provided that condition γ holds).
2. Choose a minimal set of concepts, called the 'UML heart concepts', such that all other UML concepts, called the 'UML skin concepts', can be mapped into this chosen set of concepts.
3. State the translation from the UML skin concepts into the UML heart concepts.
4. State a translation from the UML heart concepts into a precise formal framework; possible choices for such a framework are (among others) set theory, labeled state transition systems, temporal logics, streams.

The advantage by introducing the intermediate layer of UML heart concepts is that as many as possible UML concepts can be explained within UML. Such an explanation can be presented to modelers who are not experts in formal frameworks because the target language is UML, which is the language the modelers use anyway. In our comprehension, the Object Constraint Language OCL will be part of the UML heart concepts and will play a central role for the translation of UML skin concepts.

Two examples for the translation of UML skin concepts into UML heart concepts are illustrated by the two questions to follow.

2.2 QUESTION 2: Association versus Aggregation

Aggregation [is] a special form of association that specifies a whole-part relationship between the aggregate (whole) and a component part. Glossary, page B-3.

Composition [is] a form of aggregation with strong ownership and coincident lifetime as part of the whole. Parts with non-fixed multiplicity may be created after the composite itself, but once created they live and die with it (i.e. they share lifetimes).

Such parts can also be explicitly removed before the death of the composite. Glossary, page B-5.

Both kinds of aggregations define a transitive, antisymmetric relationship, i.e. the instances form a directed, non-cyclic graph. Semantics, page 2-55.

QUESTION: What is the difference between association and aggregation?

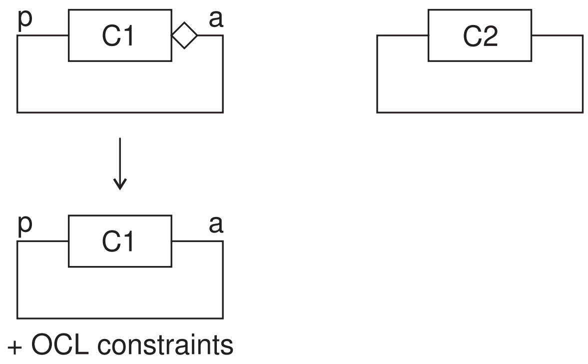

Figure 1: Association versus Aggregation

The difference is that cycles on the instance level are forbidden for aggregation relationships but are allowed for general associations: A part cannot be a part of itself. This can be expressed formally as an OCL constraint. Consider the situation in Fig. 1. The aggregation with role names `a` (for aggregate) and `p` (for part) can be translated equivalently into an association with the additional constraint:

```
C1.allInstances->forAll( c1 | not( c1.p+ -> includes(c1) ) )
```

In OCL, role names can be used as set-valued functions (we have `p: C1 -> Set(C1)`) and the function `includes` tests for set membership. In the formlula, `p+` refers to the transitive closure of `p` (the definition of the transitive closure function `p+: C1 -> Set(C1)` is also expressible in OCL). The aggregation relationship has to obey this requirement but there is no such restriction for the general association on class `C2` depicted in the right part of Fig. 1.

2.3 QUESTION 3: Unnesting of nested states

A state can be decomposed ... using or-relationships into mutually exclusive disjoint substates. ... Its substates may be refined in the same way ... Notation Guide, page 3-124.

A transition drawn to the boundary of a complex state is equivalent to a transition to its initial state ... Notation Guide, page 3-131.

A transition from a complex state indicates a transition that applies to each of the states within the state region (at any depth). Notation Guide, page 3-131.

QUESTION: Can nested states without concurrent subregions be equivalently translated into unnested states?

Yes, such a translation can be stated. One possibility for doing it is on the basis of graph transformations. This approach seems to be attractive because the language of graph transformations used for the translation directly reflects the touched part of UML. Another beenfit of this approach is its automation ability because the translation is formulated in terms of rewrite rules for the graphs underlying the statecharts.

3 Questions without Answers

3.1 QUESTION 1: Number of Classes for an Object

To show multiple classes that the object is an instance of, use a comma-separated list of classnames. These classnames must be legal for multiple classification (i.e., only one implementation class permitted, but multiple roles permitted). Notation Guide, page 3-53.

Static classification: A semantic variation of generalization in which an object may not change type or may not change role. Glossary, page B-16.

Dynamic classification: A semantic variation of generalization in which an object may change type or role. Glossary, page B-7.

Multiple classification: A semantic variation of generalization in which an object may belong directly to more than one class. Glossary, page B-11.

[Static and dynamic classification] may be used in a UML model. ... The choice affects surprisingly little a model, although the differences are important for execution. Reference Manual, page 265.

QUESTION: Are objects from two different classes interpreted always as two objects or is it possible — for example, in situations with specialization — to see them as one object?

If the one-object interpretation is chosen, how does one treat objects dynamically changing classes and what about object merging? If the two-object interpretation is chosen, how easy-to-understand is the formal interpretation? The question concerning how many classes an object may belong to is for us a key question. The answer influences many static and dynamic aspects of objects, for example, the interpretation and applicability of class diagrams and statechart diagrams.

3.2 QUESTION 2: Fork Transition versus Simple Transition to Concurrent Substates

A complex transition may have multiple source states and target states. It represents a synchronization and/or a splitting of control into concurrent threads without concurrent substates. Notation Guide, page 3-130.

A transition drawn to the boundary of a complex state is equivalent to a transition to its initial state (or to a complex transition to the initial states of each of its concurrent subregions if it is concurrent). Notation Guide, page 3-131.

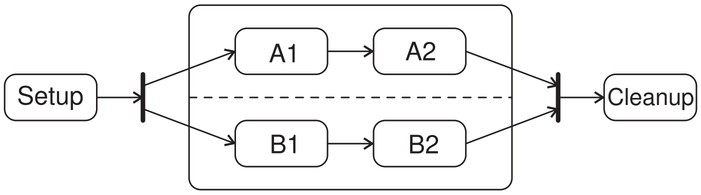

Figure 2: Example from the UML Notation Guide, Page 3-131

QUESTION: In a state machine with concurrent states, what is the difference between (1) a fork transition from an outside state to two (concurrent) inside substates (as displayed in Fig. 2) and (2) one simple transition from an outside state to two (concurrent) initial substates (as displayed in Fig. 3)? An analogous question arises for join transitions and final states.

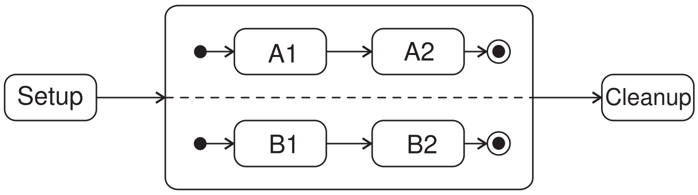

Figure 3: Figure 46 — Modified Version

There seems to be no intuitive difference between both diagrams. If the answer to the question is 'There is no difference', this might be one of the equivalence rules for statechart diagrams. But how does the general rule and the underlying derivation system look like?

4 A Collection of Questions with Answers

Can multiplicities in class diagrams be expressed exivalently with OCL constraints?

Can a ternary association in a class diagram be expressed equivalently by introducing a new class for the ternary association together with three associations and OCL constraints?

Can a qualifier in a class diagram be expressed equivalently by an association class together with OCL constraints?

Can aggregation and composition in a class diagram be expressed equivalenty by associations together with OCL constraints?

Can stubs in statechart diagrams without concurent subregions be equivalently transformed into unnested states?

Is there a formal semantics for OCL, at least for the 'safe' part of OCL?

Is it possible to express transitional constraints in OCL?

How can the existence of a ternary link be expressed in OCL?

Does flattening of collections always produce a well-defined result in OCL?

5 A Collection of Questions without Answers

Is it possible to show the history of active states in a concurrent state machine (for example the histoy in the RefMan, page 212) on a single UML diagram?

Has the read direction for an association in a class diagram any influence on the semantics?

Is it allowed to have specialization arrows in object diagrams?

Is it possible to express that three concurrent messages in a sequence diagram originate from the same object at the same time? What is the difference between (1) vertical, (2) slanted and (3) mixed (vertical/slanted) message lines?

Is it neccessary or desireable to show in a class diagram all receptions, at least if one wants an exact fitting of the class diagram to the statechart diagram?

Is it neccessary or desireable to show the complete signal hierarchy? Are signal instances objects? If the answer is yes, how can signals be treated as normal objects? If the answer is no, why does one need something which lies outside the object paradigm?

What is the interpretation of statechart/activitity diagram with mixed kinds of states (quantifier 'at least most')?

How can loops in sequence diagrams be notated?

What is the interpretation for different arrows in sequence and collaboration diagrams?

What is the interpretation of transitions from one concurrent region to another concurrent region with the same level of abstraction (as shown in Fig. 4)?

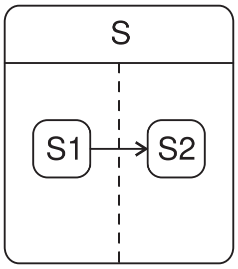

Figure 4:

Object diagrams represent the instance form for class diagrams. Represent sequence and collaboration diagrams in the same way the instance form for statechart and activity diagrams?

What is the meaning of `Integer.allInstances->size` and other touchy OCL expressions?

What is the result of `OclType.oclType`?

Is there a rule in OCL explaining how operations are selected in case of overloaded signatures or equal signatures in subtypes?

6 Relevant Literature

The original UML documents offered by the OMG [BRJ99d, BRJ99b, BRJ99c, BRJ99a] and the semi-official books on UML [BRJ98, RJB98] provide the most important information on the topic. Relevant for a UML Semantics FAQ is also work on formal foundations of UML as it is done by several research groups:

- Bremen group [GR98c, GR98b, GP98, GR98a, Gog98, RG98]
- Brighton group [KHH$^+$97, KLB$^+$97, HCH$^+$98]
- Florida/Colorado group [FBLPS97, FEL97, FB97, SLPFE98, EFLR98]
- London Group [BLM97, KLB$^+$97, LB98a, LB98b]
- Munich group [BHH$^+$97, BGH$^+$97, BGH$^+$98]
- Stockholm group [Öve98, ÖP98]
- Twente group [WB98, Wie98]

References

[BGH+97] Ruth Breu, Radu Grosu, Franz Huber, Bernhard Rumpe, and Wolfgang Schwerin. Towards a precise semantics for object-oriented modeling techniques. In Haim Kilov and Bernhard Rumpe, editors, *Proceedings ECOOP'97 Workshop on Precise Semantics for Object-Oriented Modeling Techniques*, pages 53–59. Technische Universität München, TUM-I9725, 1997.

[BGH+98] Ruth Breu, Radu Grosu, Franz Huber, Bernhard Rumpe, and Wolfgang Schwerin. Systems, views and models of UML. In Martin Schader and Axel Korthaus, editors, *The Unified Modeling Language – Technical Aspects and Applications*, pages 93–108. Physica-Verlag, Heidelberg, 1998.

[BHH+97] Ruth Breu, Ursula Hinkel, Christoph Hofmann, Cornel Klein, Barbara Paech, Bernhard Rumpe, and Veronika Thurner. Towards a formalization of the unified modeling language. In Mehmet Aksit and Satoshi Matsuoka, editors, *ECOOP'97 – Object-Oriented Programming, 11th European Conference*, volume 1241 of *LNCS*. Springer, 1997.

[BLM97] Juan Bicarregui, Kevin Lano, and Tom Maibaum. Formalising object-oriented models in the object calculus. In Haim Kilov and Bernhard Rumpe, editors, *Proceedings ECOOP'97 Workshop on Precise Semantics for Object-Oriented Modeling Techniques*, pages 45–51. Technische Universität München, TUM-I9725, 1997.

[BRJ98] Grady Booch, James Rumbaugh, and Ivar Jacobson. *The Unified Modeling Language User Guide*. Addison-Wesley, 1998.

[BRJ99a] G. Booch, J. Rumbaugh, and I. Jacobsen, editors. *Object Constraint Language OCL (Version 1.3a1)*. OMG, 1999. Version on CD delivered with UML Reference Manual.

[BRJ99b] G. Booch, J. Rumbaugh, and I. Jacobsen, editors. *UML Notation Guide (Version 1.3a1)*. OMG, 1999. Version on CD delivered with UML Reference Manual.

[BRJ99c] G. Booch, J. Rumbaugh, and I. Jacobsen, editors. *UML Semantics (Version 1.3a1)*. OMG, 1999. Version on CD delivered with UML Reference Manual.

[BRJ99d] G. Booch, J. Rumbaugh, and I. Jacobsen, editors. *Unified Modeling Language (Version 1.3a1)*. OMG, 1999. Version on CD delivered with UML Reference Manual.

[EFLR98] Andy Evans, Robert France, Kevin Lano, and Bernhard Rumpe. Developing the UML as a formal modelling notation. In Pierre-Alain Muller and Jean Bézivin, editors, *Proceedings of UML'98 International Workshop, Mulhouse, France, June 3 - 4, 1998*, pages 297–307. ESSAIM, Mulhouse, France, 1998.

[FB97] Robert France and Jean-Michel Bruel. Integrated informal object-oriented and formal modeling techniques. In Haim Kilov and Bernhard Rumpe, editors, *Proceedings ECOOP'97 Workshop on Precise Semantics for*

Object-Oriented Modeling Techniques, pages 83–89. Technische Universität München, TUM-I9725, 1997.

[FBLPS97] Robert France, Jean-Michel Bruel, Maria Larrondo-Petrie, and M. Shroff. Exploring the semantics of UML type structures with Z. In H. Bowman and J. Derrick, editors, *Proc. 2nd IFIP Conf. Formal Methods for Open Object-Based Distributed Systems (FMOODS'97)*. Chapman and Hall, London, 1997.

[FEL97] Robert France, Andy Evans, and Kevin Lano. The UML as a formal modeling notation. In Haim Kilov, Bernhard Rumpe, and Ian Simmonds, editors, *Proceedings OOPSLA'97 Workshop on Object-oriented Behavioral Semantics*, pages 75–81. Technische Universität München, TUM-I9737, 1997.

[Gog98] Martin Gogolla. UML for the impatient. Research Report 3/98, Universität Bremen, 1998.

[GP98] Martin Gogolla and Francesco Parisi Presicce. State diagrams in UML: A formal semantics using graph transformations. In Manfred Broy, Derek Coleman, Tom S. E. Maibaum, and Bernhard Rumpe, editors, *Proceedings PSMT'98 Workshop on Precise Semantics for Modeling Techniques.* Technische Universität München, TUM-I9803, 1998.

[GR98a] Martin Gogolla and Mark Richters. Equivalence rules for UML class diagrams. In Pierre-Alain Muller and Jean Bézivin, editors, *Proceedings of UML'98 International Workshop, Mulhouse, France, June 3 - 4, 1998*, pages 87–96. ESSAIM, Mulhouse, France, 1998.

[GR98b] Martin Gogolla and Mark Richters. On combing semi-formal and formal object specification techniques. In Francesco Parisi-Presice, editor, *Recent trends in algebraic development techniques: 12th international workshop, WADT'97, Tarquinia, Italy, June 3–7, 1997: selected papers*, volume 1376 of *LNCS*. Springer, 1998.

[GR98c] Martin Gogolla and Mark Richters. On constraints and queries in UML. In Martin Schader and Axel Korthaus, editors, *The Unified Modeling Language – Technical Aspects and Applications*, pages 109–121. Physica-Verlag, Heidelberg, 1998.

[HCH+98] Ali Hamie, Franco Civello, John Howse, Stuart Kent, and Richard Mitchell. Reflections on the object constraint language. In Pierre-Alain Muller and Jean Bézivin, editors, *Proceedings of UML'98 International Workshop, Mulhouse, France, June 3 - 4, 1998*, pages 137–145. ESSAIM, Mulhouse, France, 1998.

[KHH+97] Stuart Kent, Ali Hamie, John Howse, Franco Civello, and Richard Mitchell. Semantics through pictures: Towards a diagrammatic semantics for OO modeling notations. In Haim Kilov and Bernhard Rumpe, editors, *Proceedings ECOOP'97 Workshop on Precise Semantics for Object-Oriented Modeling Techniques*, pages 91–100. Technische Universität München, TUM-I9725, 1997.

[KLB+97] Stuart Kent, Kevin Lano, Juan Bicarregui, Ali Hamie, and John Howse. Component composition in business and system modeling. In Haim Kilov, Bernhard Rumpe, and Ian Simmonds, editors, *Proceedings OOPSLA'97 Workshop on Object-oriented Behavioral Semantics*, pages 91–97. Technische Universität München, TUM-I9737, 1997.

[LB98a] Kevin Lano and Juan Bicarregui. Formalising the UML in structured temporal theories. In Haim Kilov and Bernhard Rumpe, editors, *Proceedings Second ECOOP Workshop on Precise Behavioral Semantics (with an Emphasis on OO Business Specifications)*, pages 105–121. Technische Universität München, TUM-I9813, 1998.

[LB98b] Kevin Lano and Juan Bicarregui. Semantics and transformations for UML models. In Pierre-Alain Muller and Jean Bézivin, editors, *Proceedings of UML'98 International Workshop, Mulhouse, France, June 3 - 4, 1998*, pages 97–106. ESSAIM, Mulhouse, France, 1998.

[ÖP98] Gunnar Övergaard and Karin Palmkvist. A formal approach to use cases and their relationships. In Pierre-Alain Muller and Jean Bézivin, editors, *Proceedings of UML'98 International Workshop, Mulhouse, France, June 3 - 4, 1998*, pages 309–317. ESSAIM, Mulhouse, France, 1998.

[Öve98] Gunnar Övergaard. A formal approach to relationships in the Unified Modeling Language. In Manfred Broy, Derek Coleman, Tom S. E. Maibaum, and Bernhard Rumpe, editors, *Proceedings PSMT'98 Workshop on Precise Semantics for Modeling Techniques*. Technische Universität München, TUM-I9803, 1998.

[RG98] Mark Richters and Martin Gogolla. On formalizing the UML object constraint language OCL. In Tok Wang Ling, Sudha Ram, and Mong Li Lee, editors, *Proc. 17th Int. Conf. Conceptual Modeling (ER'98)*. Springer, Berlin, LNCS 1507, 1998.

[RJB98] James Rumbaugh, Ivar Jacobson, and Grady Booch. *The Unified Modeling Language Reference Manual*. Addison-Wesley, 1998.

[SLPFE98] Monika Saksena, Maria Larrondo-Petrie, Robert France, and Matthew Evett. Extending aggregation constructs in UML. In Pierre-Alain Muller and Jean Bézivin, editors, *Proceedings of UML'98 International Workshop, Mulhouse, France, June 3 - 4, 1998*, pages 273–280. ESSAIM, Mulhouse, France, 1998.

[WB98] Roel Wieringa and Jan Broersen. A minimal transition system semantics for lightweight class- and behavior diagrams. In Manfred Broy, Derek Coleman, Tom S. E. Maibaum, and Bernhard Rumpe, editors, *Proceedings PSMT'98 Workshop on Precise Semantics for Modeling Techniques*. Technische Universität München, TUM-I9803, 1998.

[Wie98] Roel Wieringa. Formalizing the UML in a systems engineering approach. In Haim Kilov and Bernhard Rumpe, editors, *Proceedings Second ECOOP Workshop on Precise Behavioral Semantics (with an Emphasis on OO Business Specifications)*, pages 254–266. Technische Universität München, TUM-I9813, 1998.