Modelling Of Aspect Weaving

Wisam Al Abed

Modeling Aspect-Oriented Framework: Handling Aspect Reusability, Dependencies, Variabilities and Conflicts

Jörg Kienzle, Jacques Klein, and Nicolas Guelfi

Agenda

- Present by Example
  - The Simple Approach
  - The Aspect Oriented Approach
- Summarize and Conclude
- Questions

Our Transaction Example

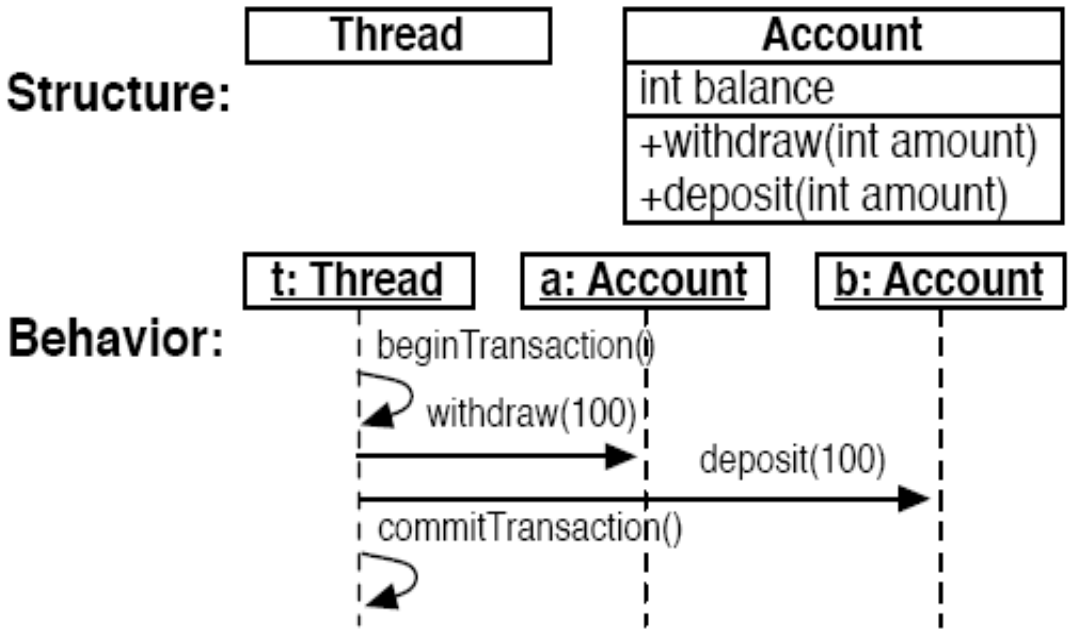

Fig. 5. A Simple Application Model

Transaction Product Line

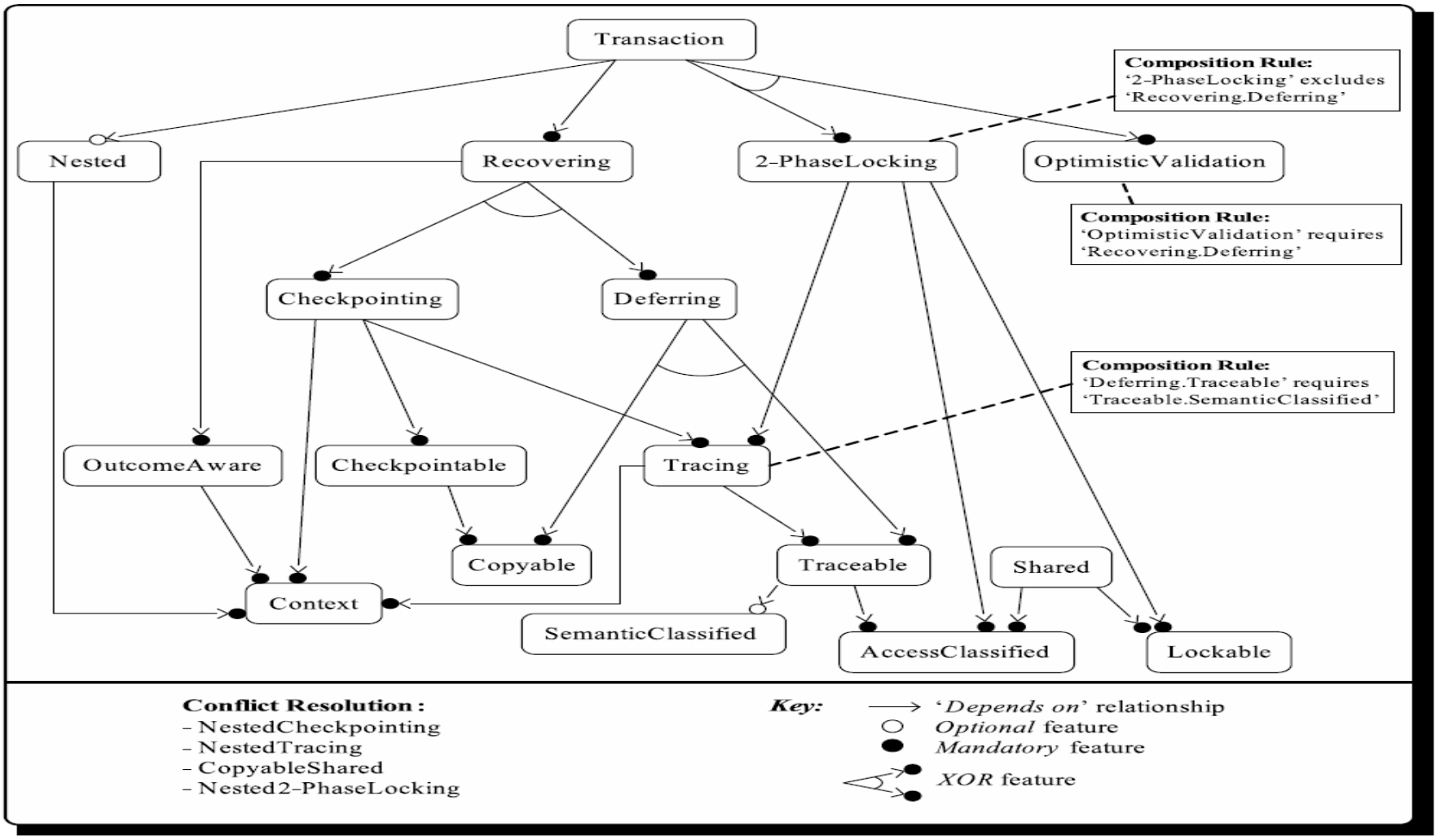

Fig. 16. Our Product Line of Transaction

Flat Transaction

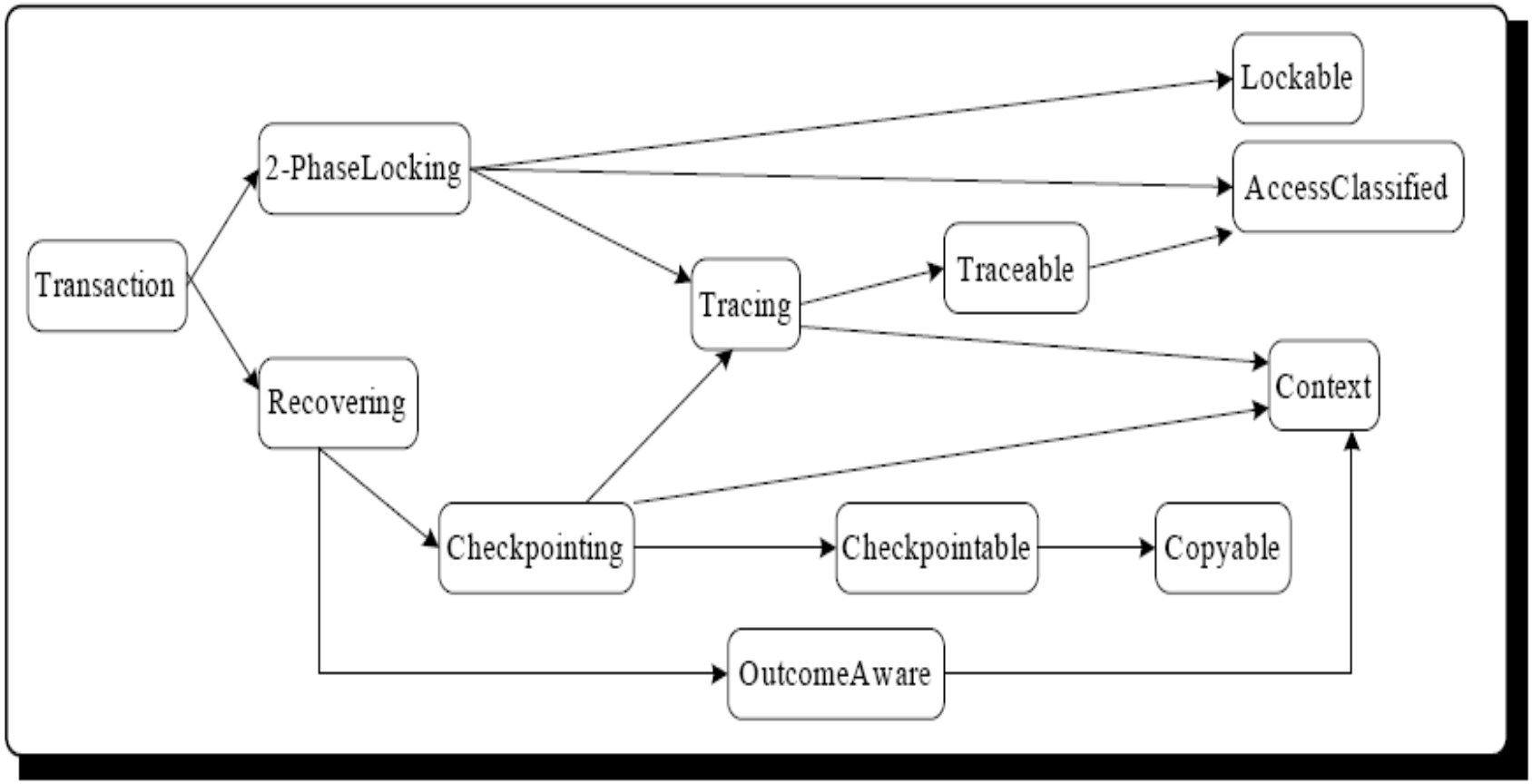

Fig. 18. Aspect Dependency chain for a Flat Transaction

Class Diagram Weaving

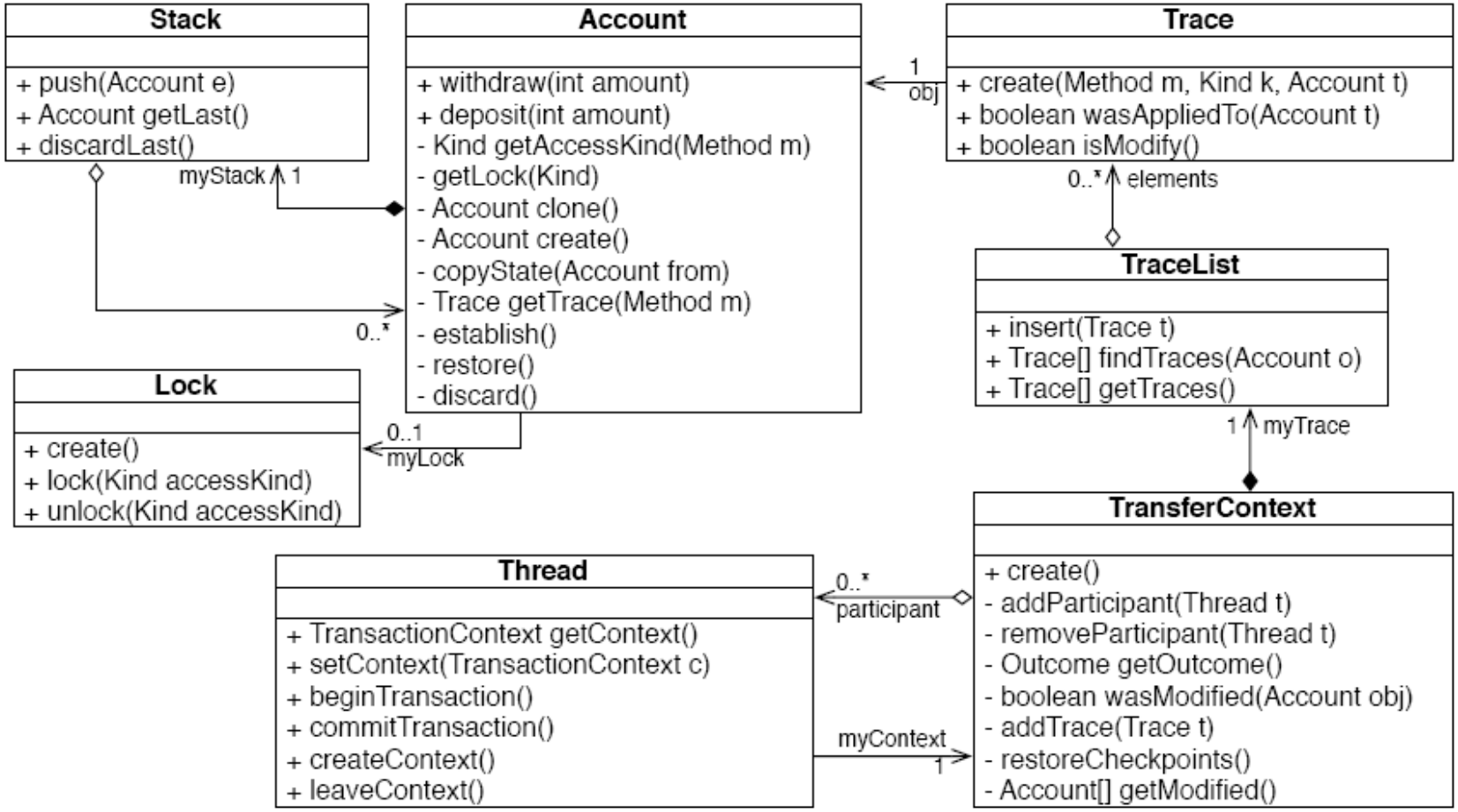

Fig. 19. Structure of Woven Application Model

Sequence Diagram Weaving

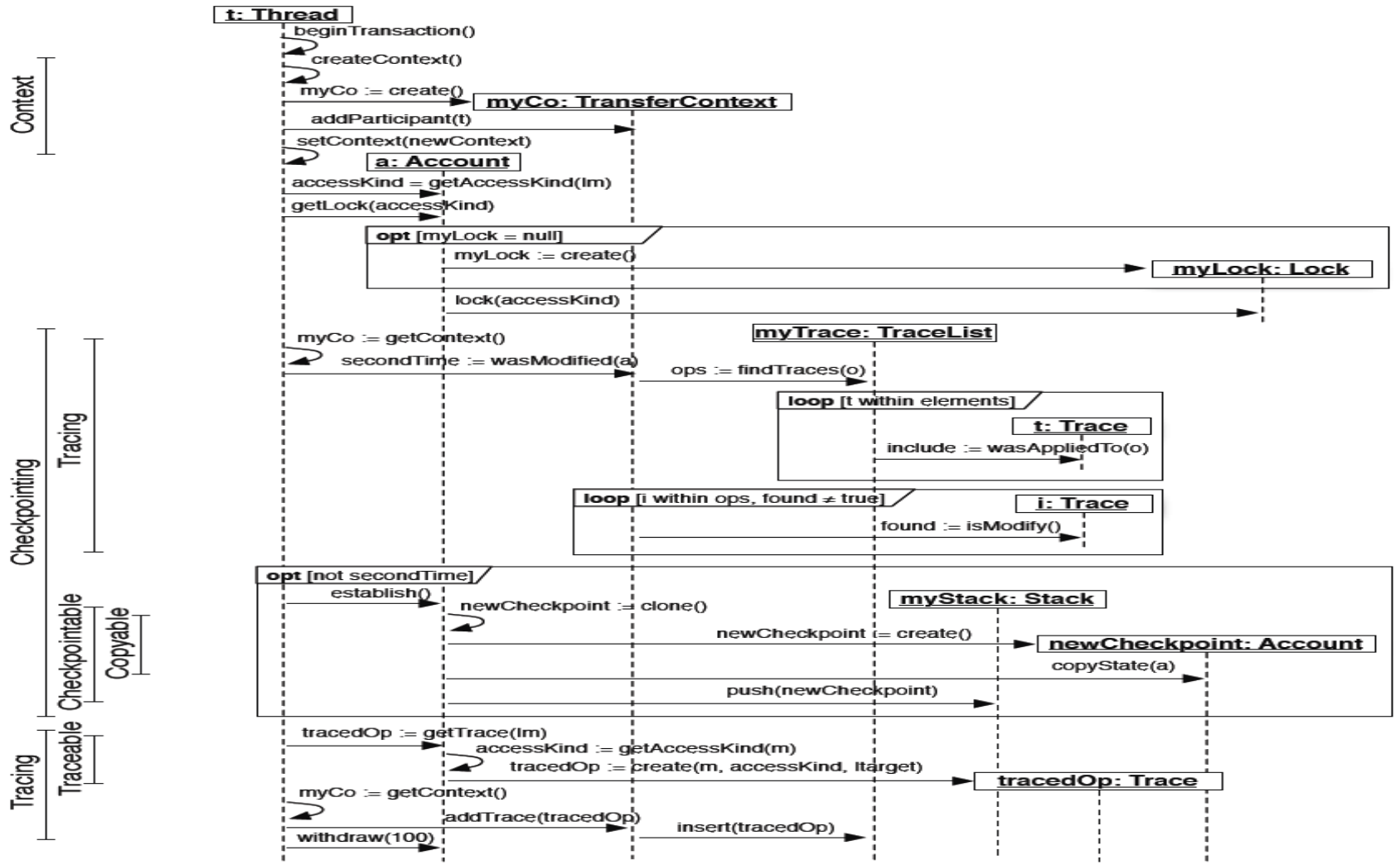

Fig. 20. Behavior of Woven Application Model

Transaction Product Line

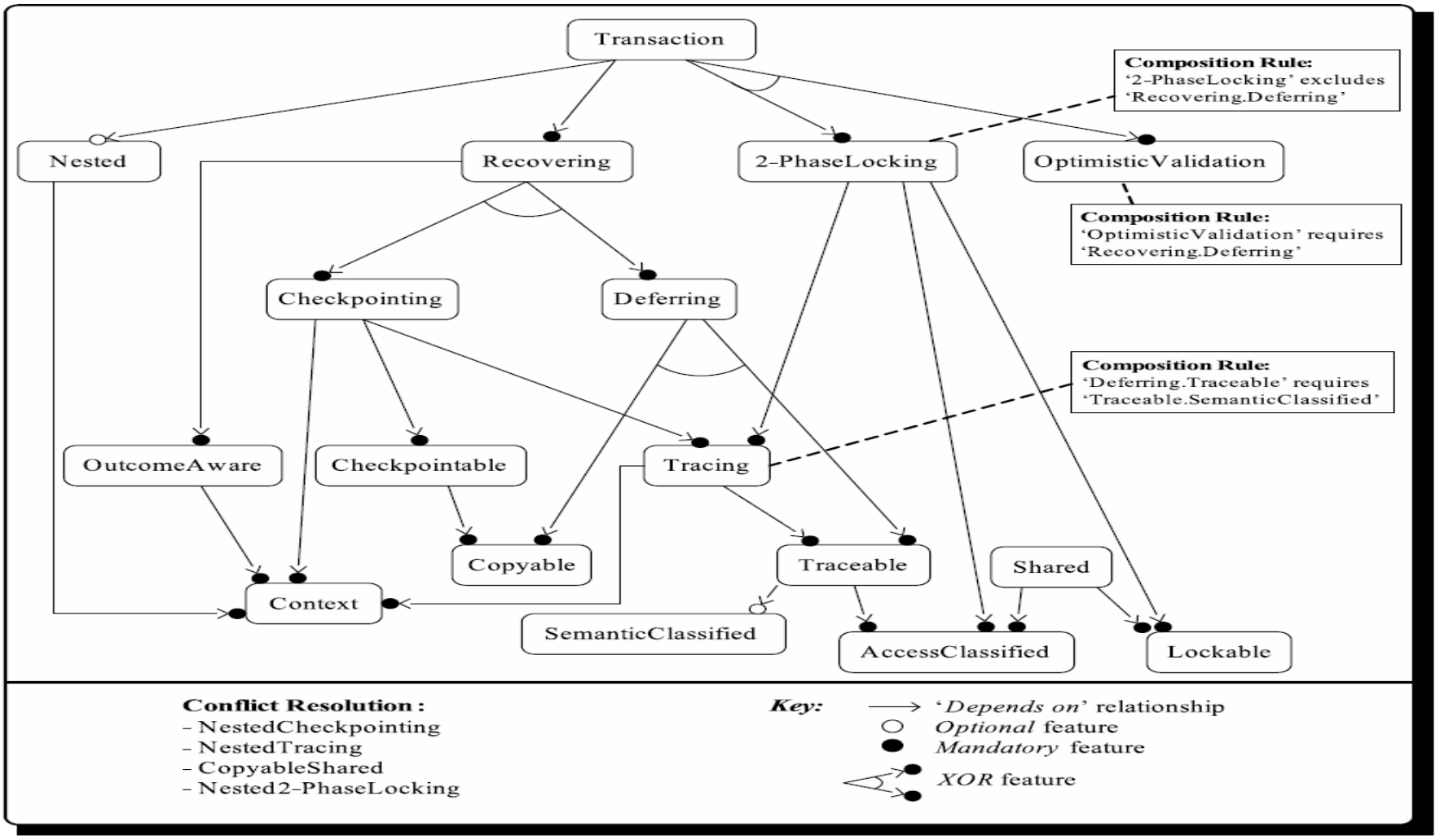

Fig. 16. Our Product Line of Transaction

Aspect Oriented Modelling

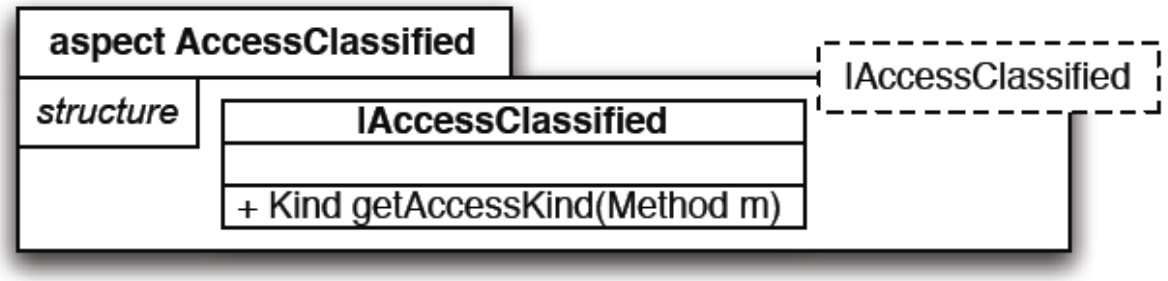

Fig. 6. The *AccessClassified* Aspect

we use UML template parameters to identify within an aspect model which modeling elements are generic. Reusing an aspect model involves binding the aspect model's template parameters to target model-specific elements.

Aspect Oriented Modelling

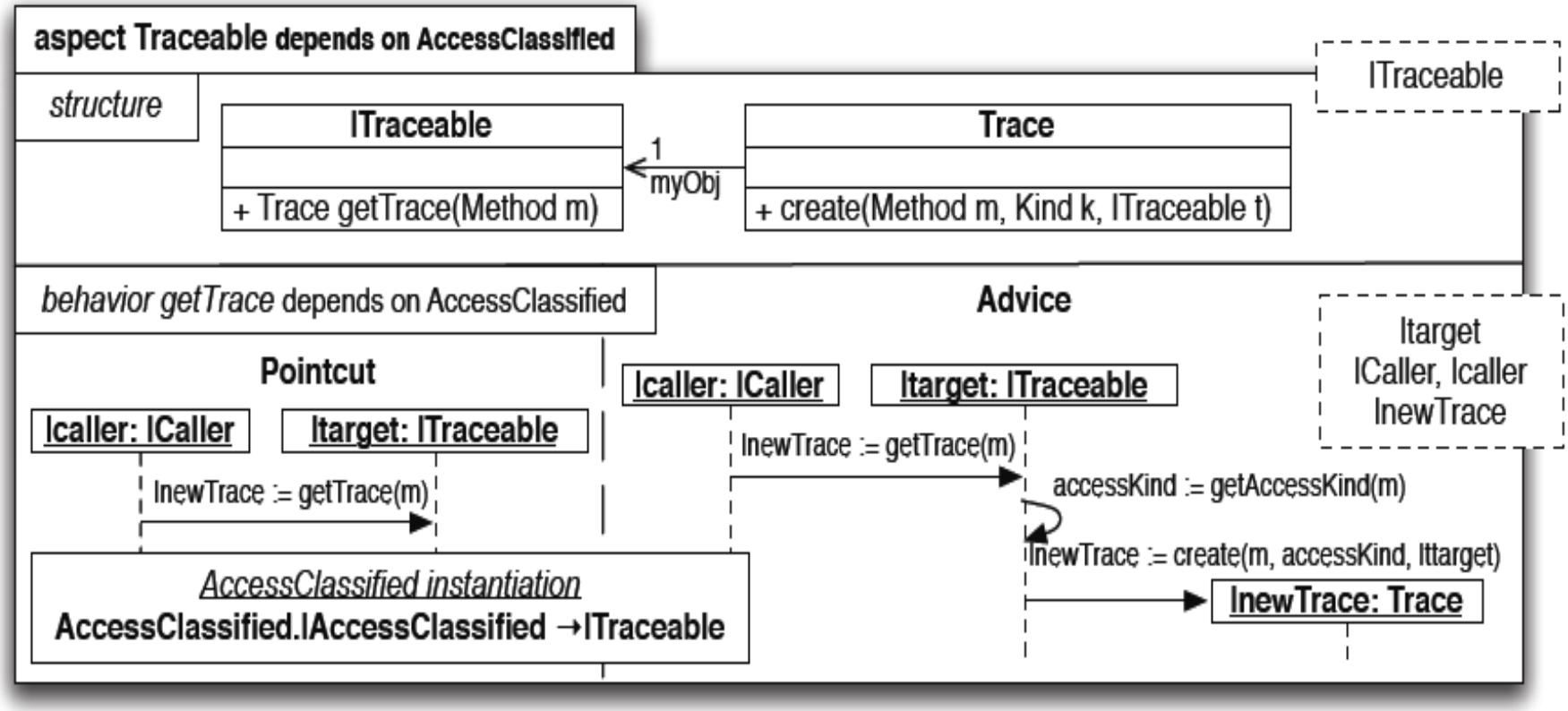

Fig. 9. The *Traceable* Aspect depends on *AccessClassified*

Flat Transaction

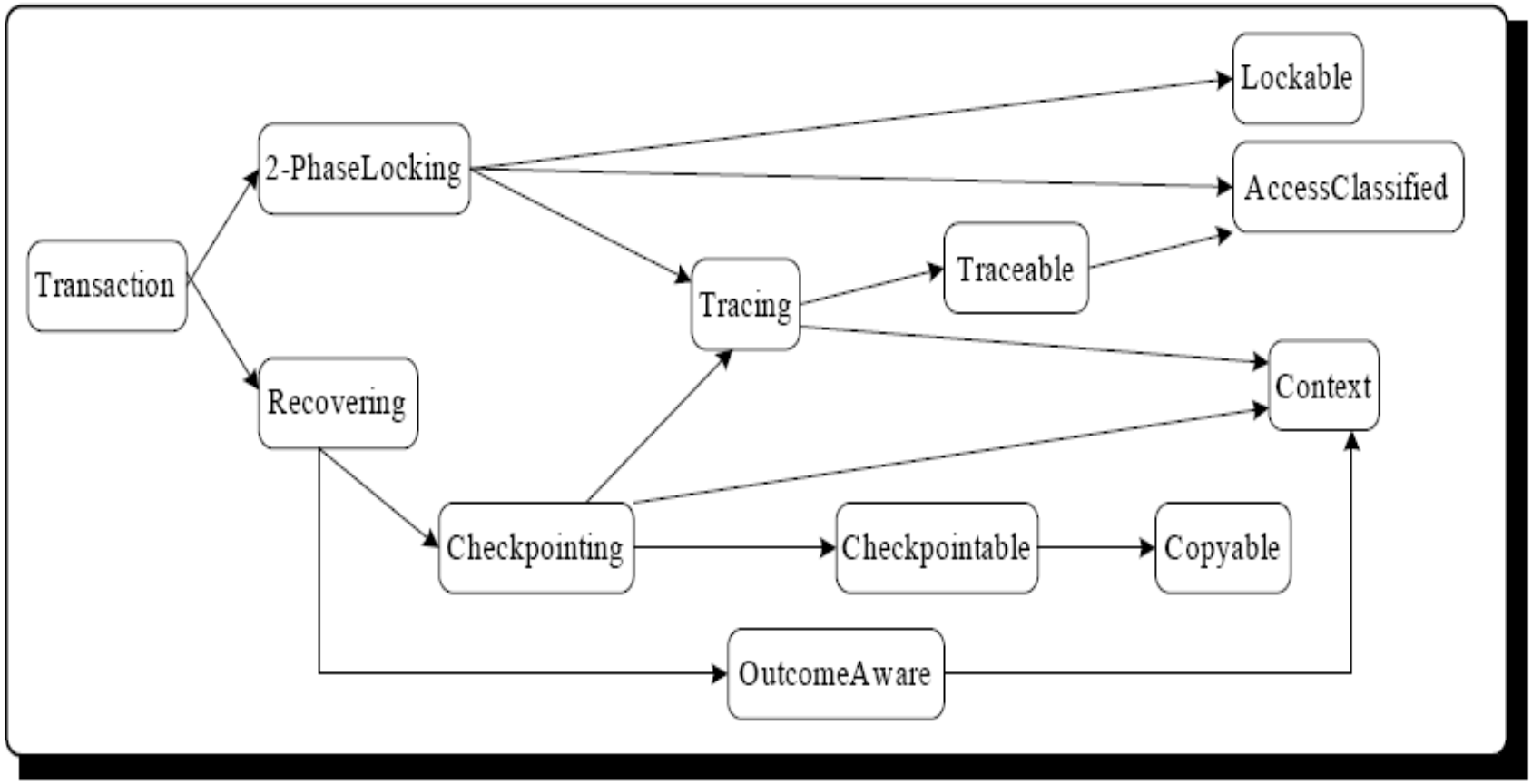

Fig. 18. Aspect Dependency chain for a Flat Transaction

Transaction Aspect

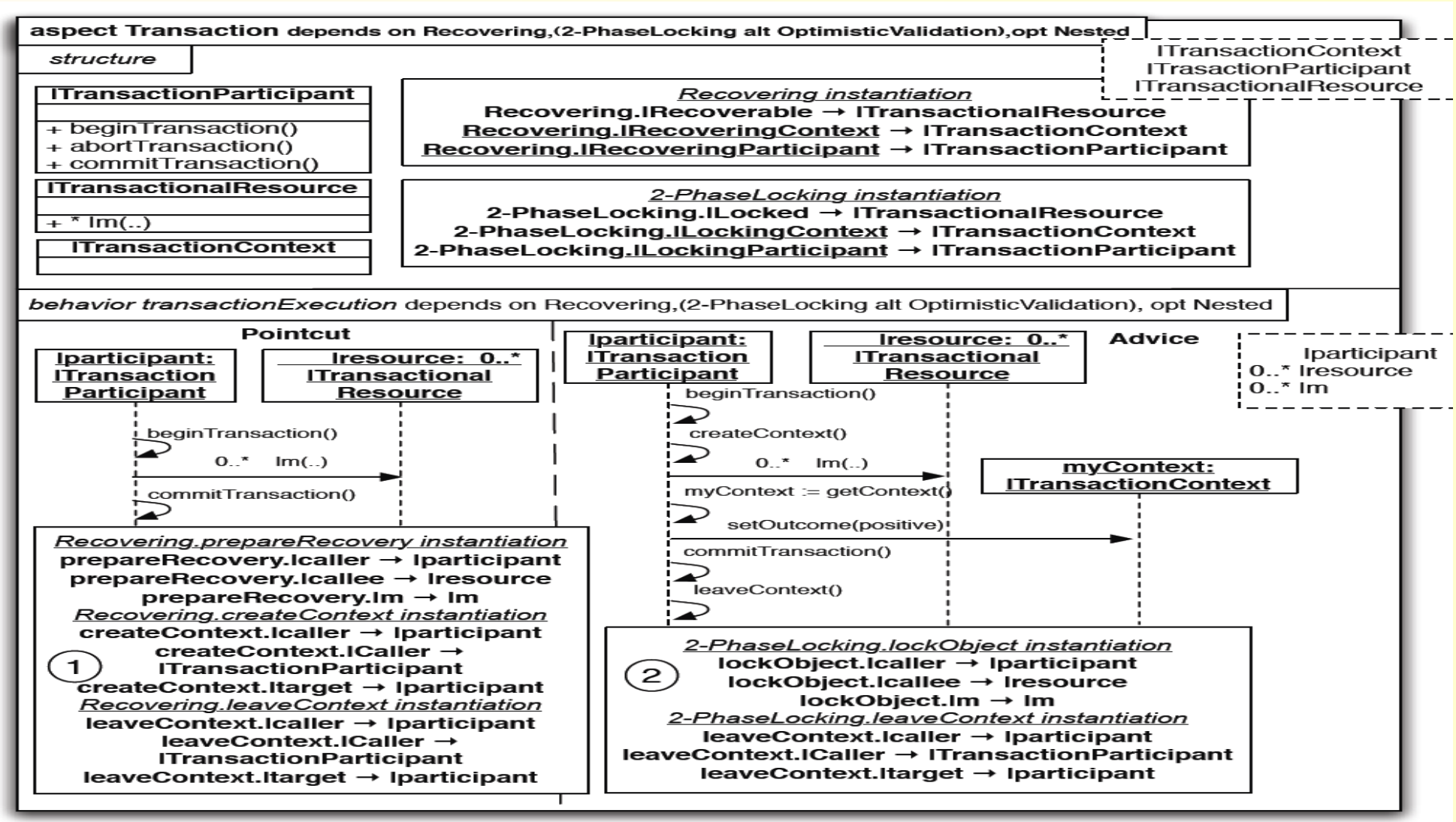

Fig. 17. The Transaction Aspect

Summarize

- Template Parameters and Binding / Instantiation:
  - Template parameters are used in a reusable aspect model as generic placeholders for structure or behavior expected by the target model elements. Instantiating a reusable aspect model binds the template parameters to target model elements, thus creating a context-specific aspect model.
- Inter-Aspect Dependency Declaration:
  - If aspect A depends on a reusable aspect B, A must provide instantiation directives for B.

Summarize

- Template Parameter-Preserving Aspect-Aspect Weaving:
  - It should be possible to weave an aspect model with another aspect model. This is mandatory to allow aspects to be reused within other aspects. The weaver must also preserve any template parameters that are not bound to specific model elements during the weaving.
- Late Dependency Resolution:
  - When a reusable aspect model A is instantiated and woven with a target model, the weaver must first recursively instantiate and weave all aspects that A depends on with A (to create an independent model of A). Instantiation and weaving due to dependencies is performed only during the final target model creation. Independent intermediate aspect models are never exposed.

Q&A

ASK THE EXPERT???