



On the Importance of Inter-Scenario Relationships

Francis Bordeleau
Jean-Pierre Corriveau
School of Computer Science
Carleton University
francis@scs.carleton.ca

Outline

- **Motivation**
- **Problem**
- **Proposed approach**
- **Inter-scenario relationships**
- **Related work**
- **Summary and future work**

Motivation

- **Scenario models and hierarchical state machine models play central roles in current object-oriented modeling techniques**
- **They provide two orthogonal views of real-time systems**
 - **Scenario models describe system behavior as sequences of responsibilities (or messages) that need to be executed by components in order to achieve overall system objectives (i.e. one scenario, many components)**
 - **Hierarchical state machine models describe complete component behavior in terms of states and transitions (i.e. one components, many scenarios)**

Problem

One of the most crucial and complex phases of complex object-oriented system design lies in the transition that is required to go from system behavior (defined by means of scenario models) to component behavior (described by means of communicating hierarchical state machine models)

Complexity Factors

Among the factors that contribute to this complexity are:

- Large number of scenarios**
- Concurrency and interactions between scenarios**
- Normal scenarios (main or primary scenarios) vs. alternatives (secondary scenarios) and exception scenarios**
- Scenarios related to different aspects of the system (e.g. operational, control, maintenance, error handling, etc)**
- Unpredictability of external events**
- Component must be design to facilitate maintainability, extensibility**
- Nonfunctional requirements (e.g. performance and robustness)**

From Scenarios to Hierarchical State Machine in UML

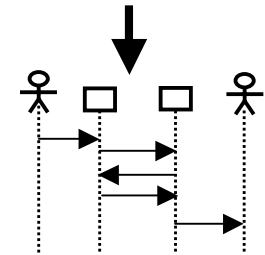
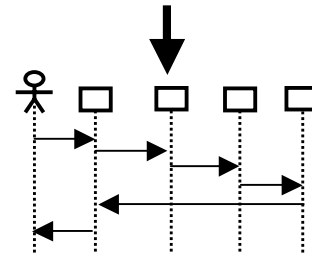
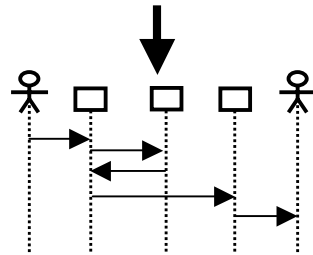
Use Cases

Use Case 1

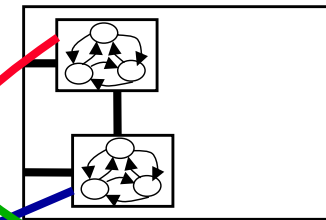
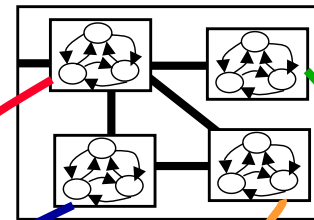
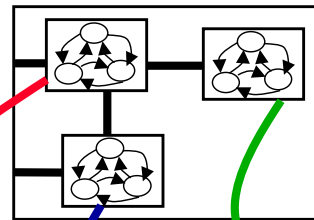
Use Case 2

Use Case 3

Sequence Diagrams

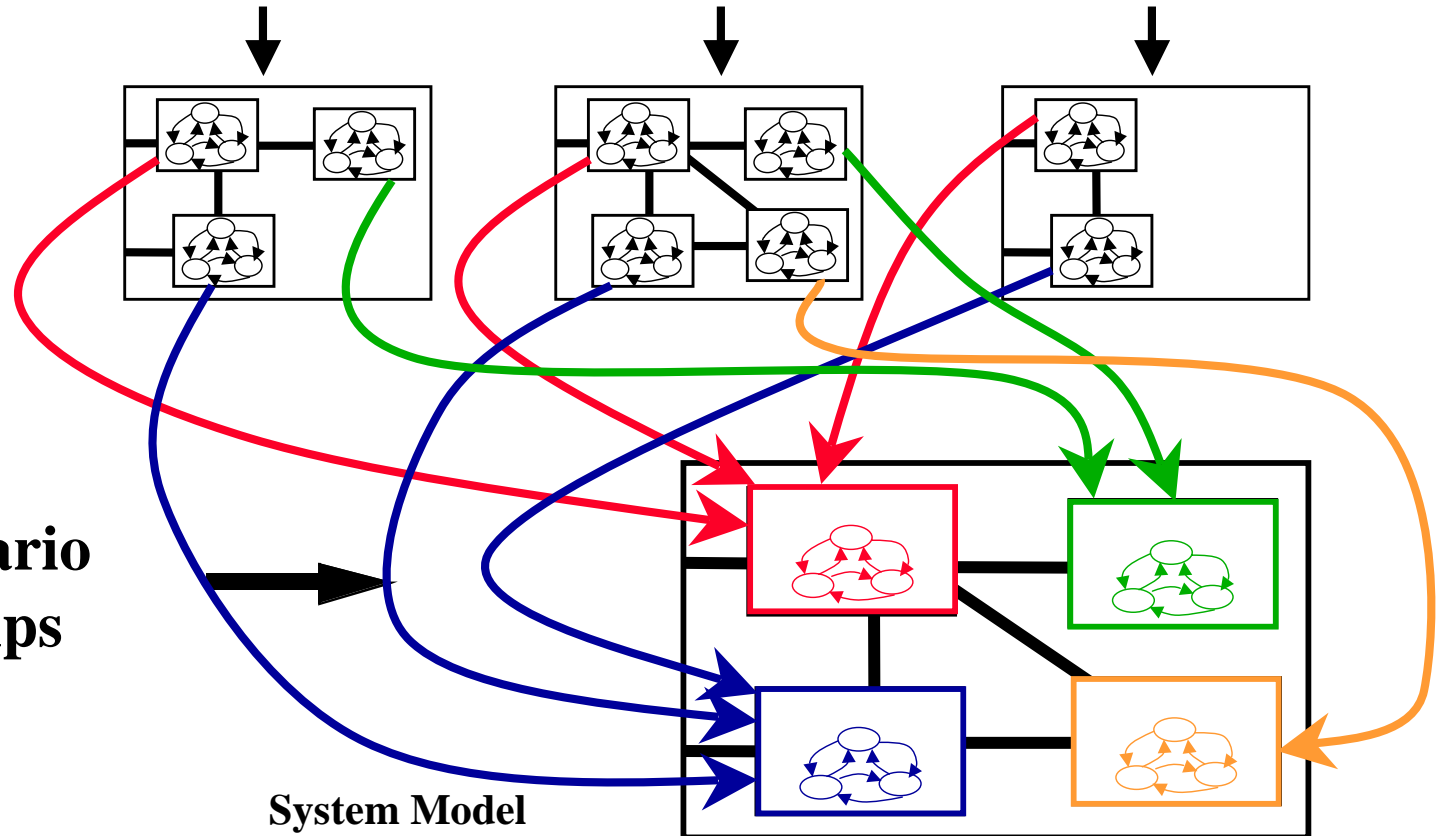


Role Models
Capsule diagrams
+ Statechart



Inter-Scenario Relationships

System Model
(Capsule diagrams + Statechart)



Proposed ITU Development Process

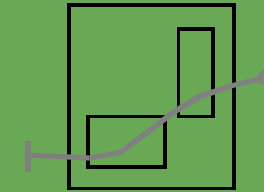
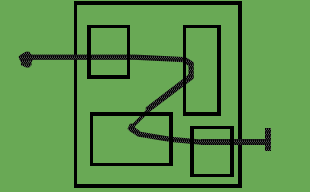
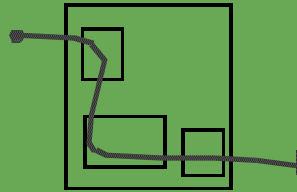
Requirements

Use Case 1

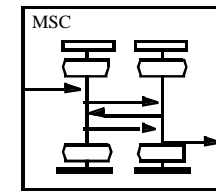
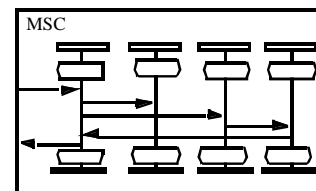
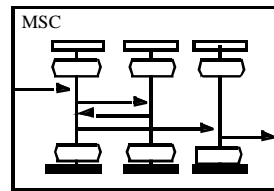
Use Case 2

Use Case 3

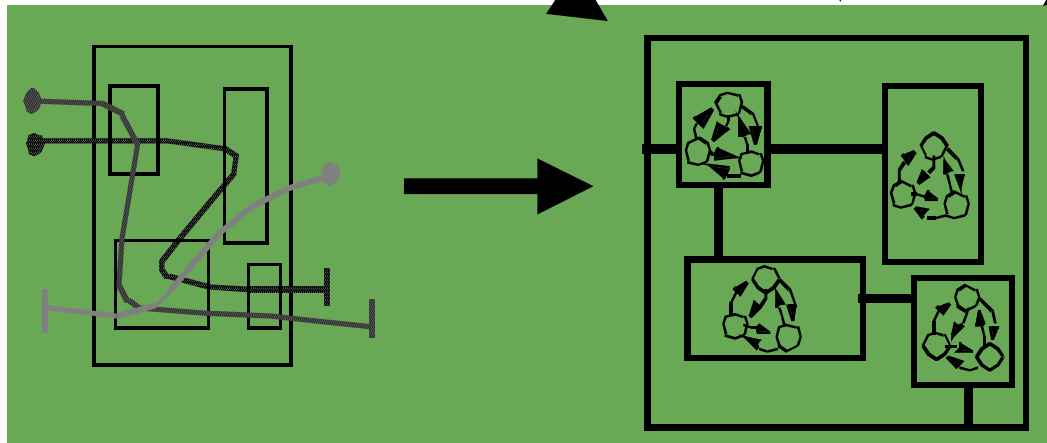
Stage 1
UCM



Stage 2
MSC



Stage 3
SDL



Inter-Scenario Relationships

The specification and understanding of inter-scenario relationships constitutes a fundamental aspect of complex system design

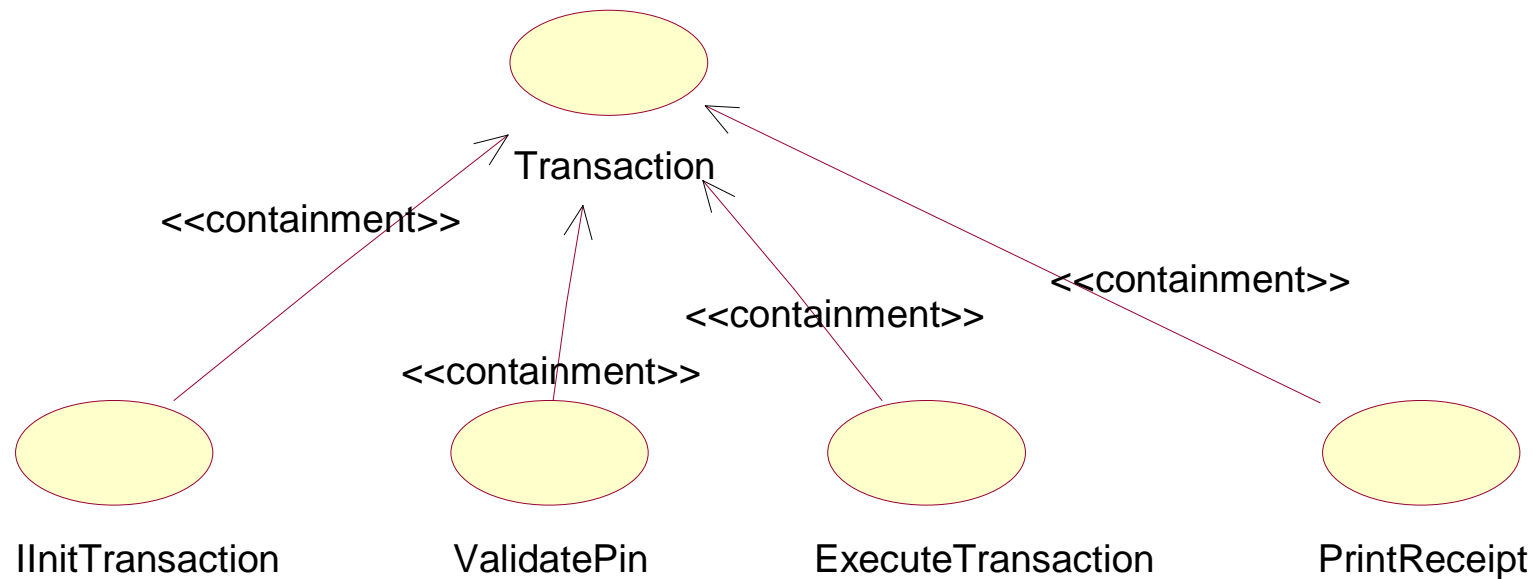
Use case relationships are not sufficient

We identify four types of inter-scenario relationships:

- **Containment Dependency**
- **Alternative Dependency**
- **Temporal Dependency**
- **Functional (or Logical) Dependency**

Containment Dependency

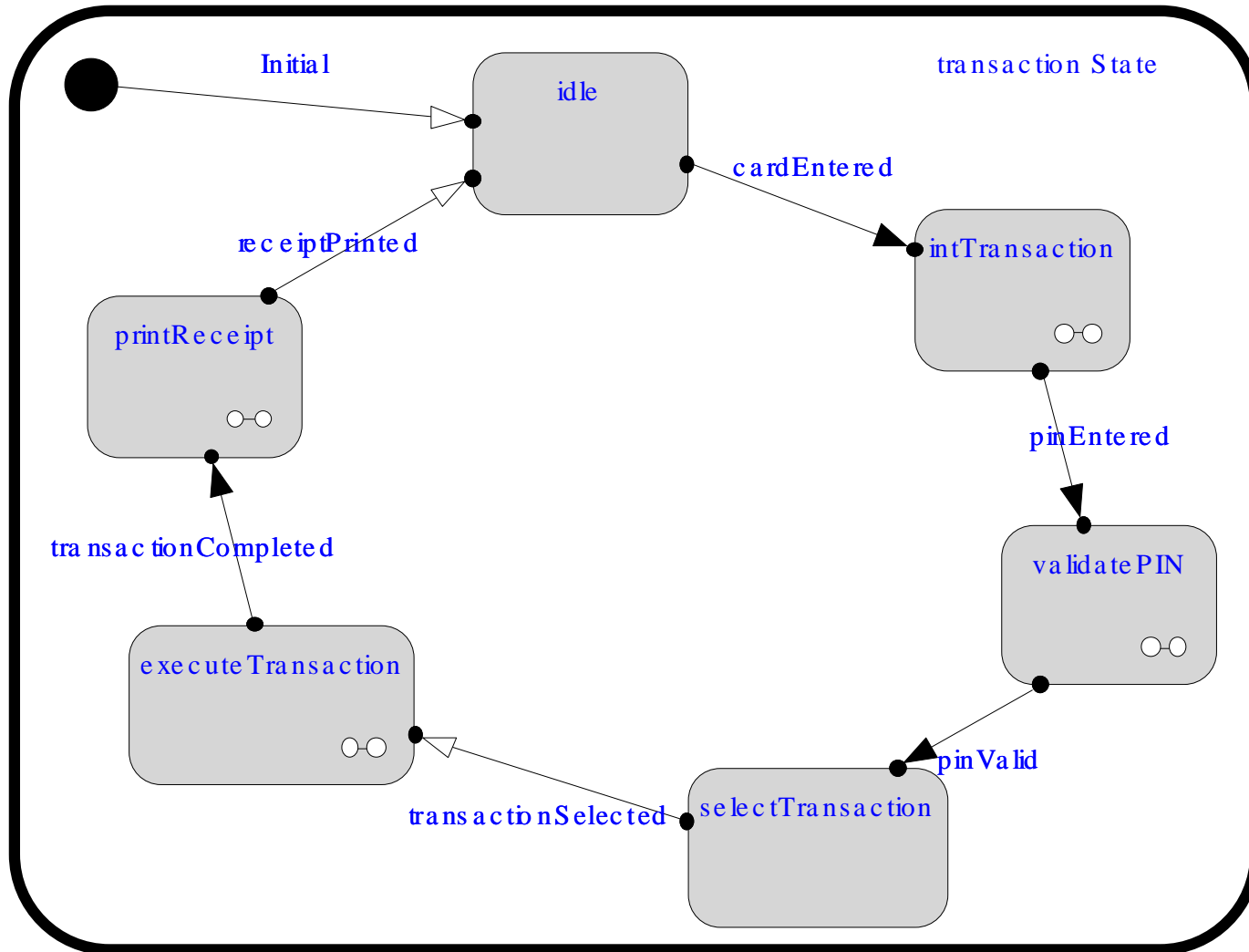
A containment dependency exists between a scenario S2 and a scenario S1, if S2 is used in the description of S1



Examples include:

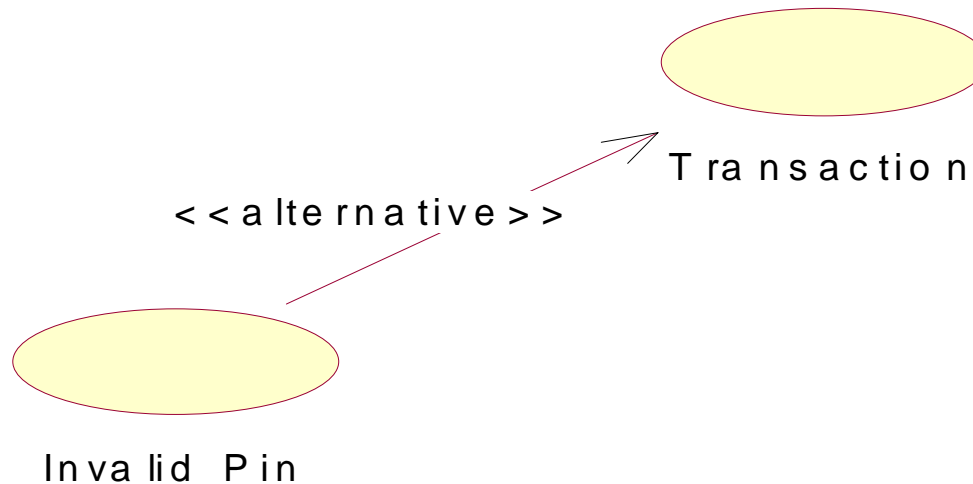
- **Include** relationship in Use Cases
- **Supported in UML activity diagrams**
- **MSC references** in Message Sequence Charts
- **Not supported** in UML interaction diagrams
- **Stubs** in Use Case Maps

Containment Dependency



Alternative Dependency

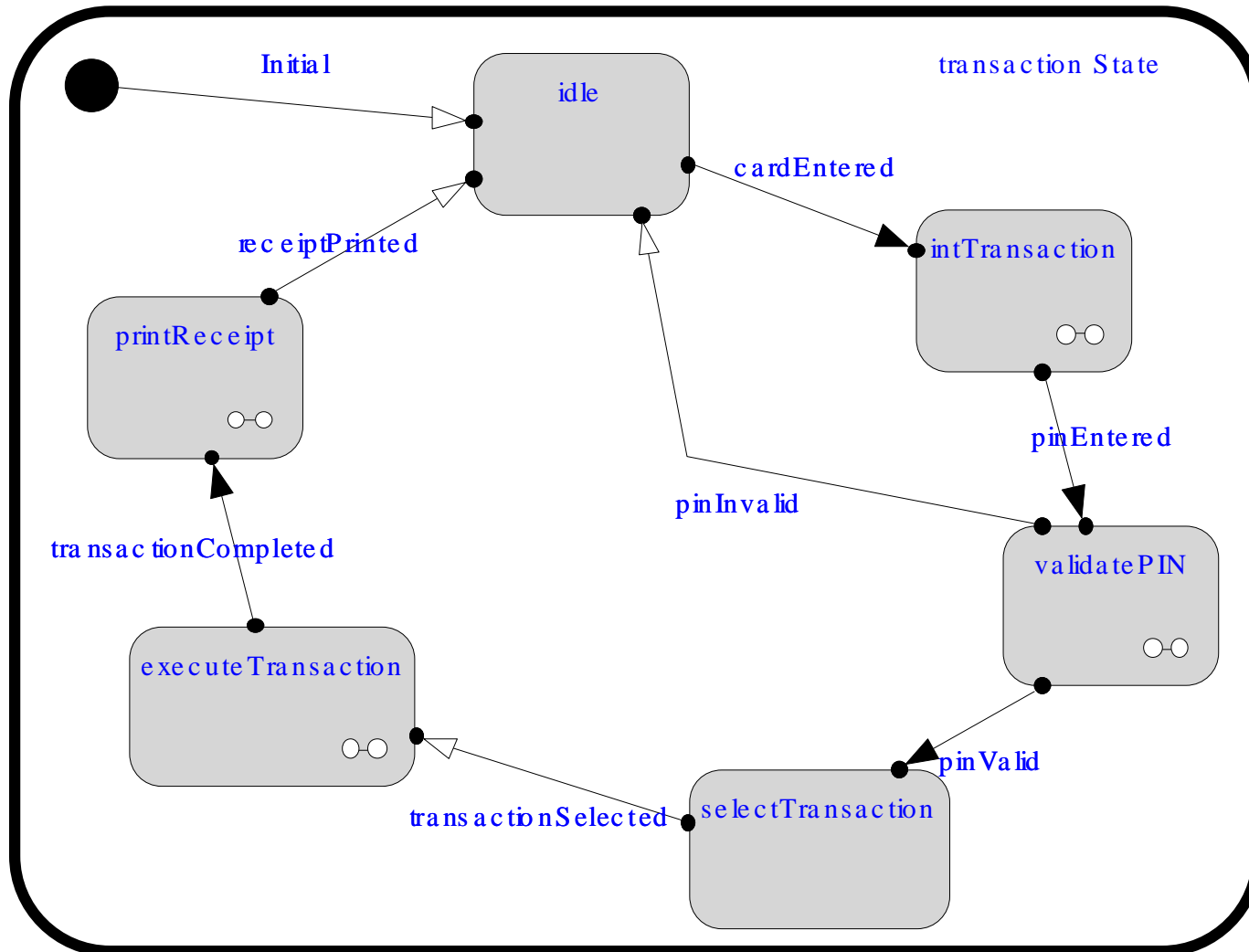
An alternative dependency exists between a scenario S2 and a scenario S1, if S2 is an alternative to S1



Examples include:

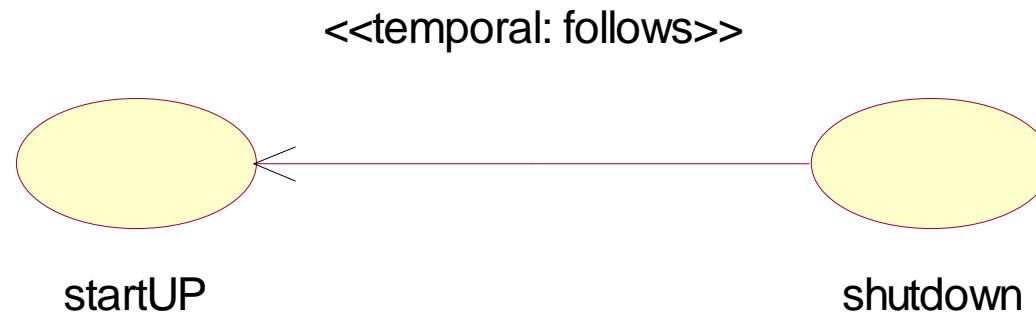
- **Primary and secondary scenarios** in Use Cases
- **Alternative constructs** in Message Sequence Charts
- **Numbering scheme** in UML collaboration diagrams
- **Branching** in UML sequence diagrams
- **Supported** in UML activity diagrams
- **Path sharing a common start point** in Use Case Maps

Alternative Dependency



Temporal Dependency

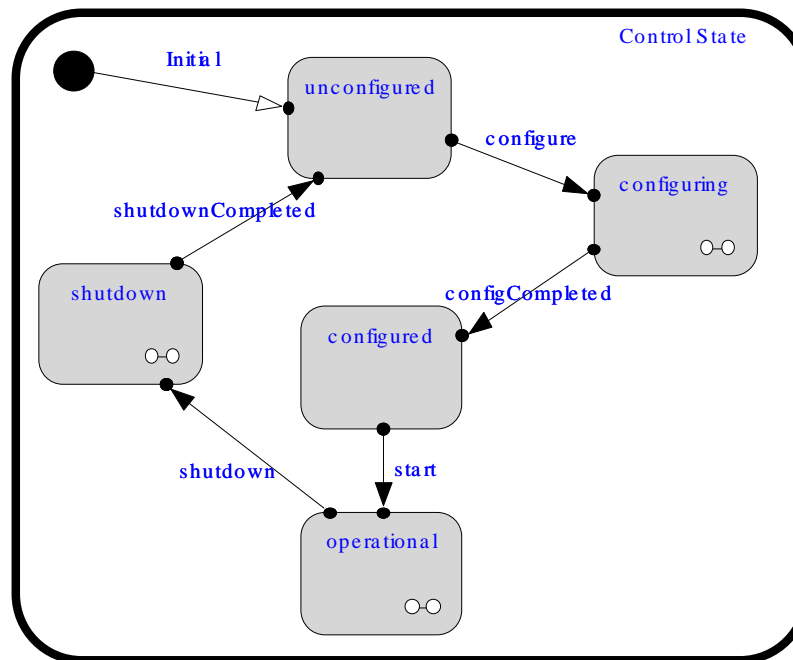
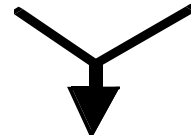
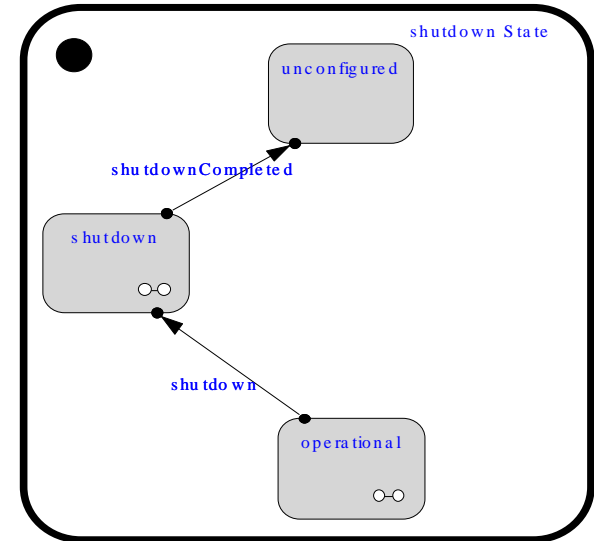
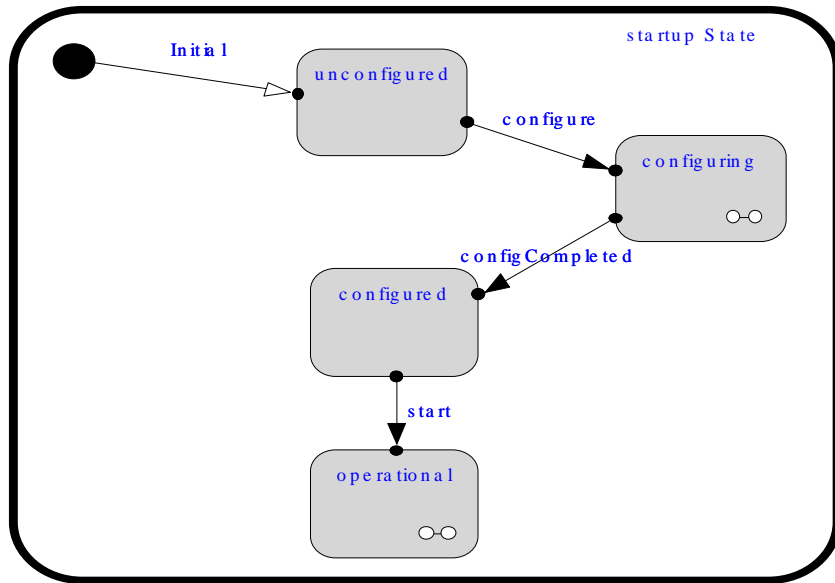
Capture different types of temporal relationships that may exist between scenarios (*e.g.*, one scenario *excludes*, *waits for*, *aborts*, *rendezvous* or *joins* another, *concurrent*, *mutually exclusive*, etc.)



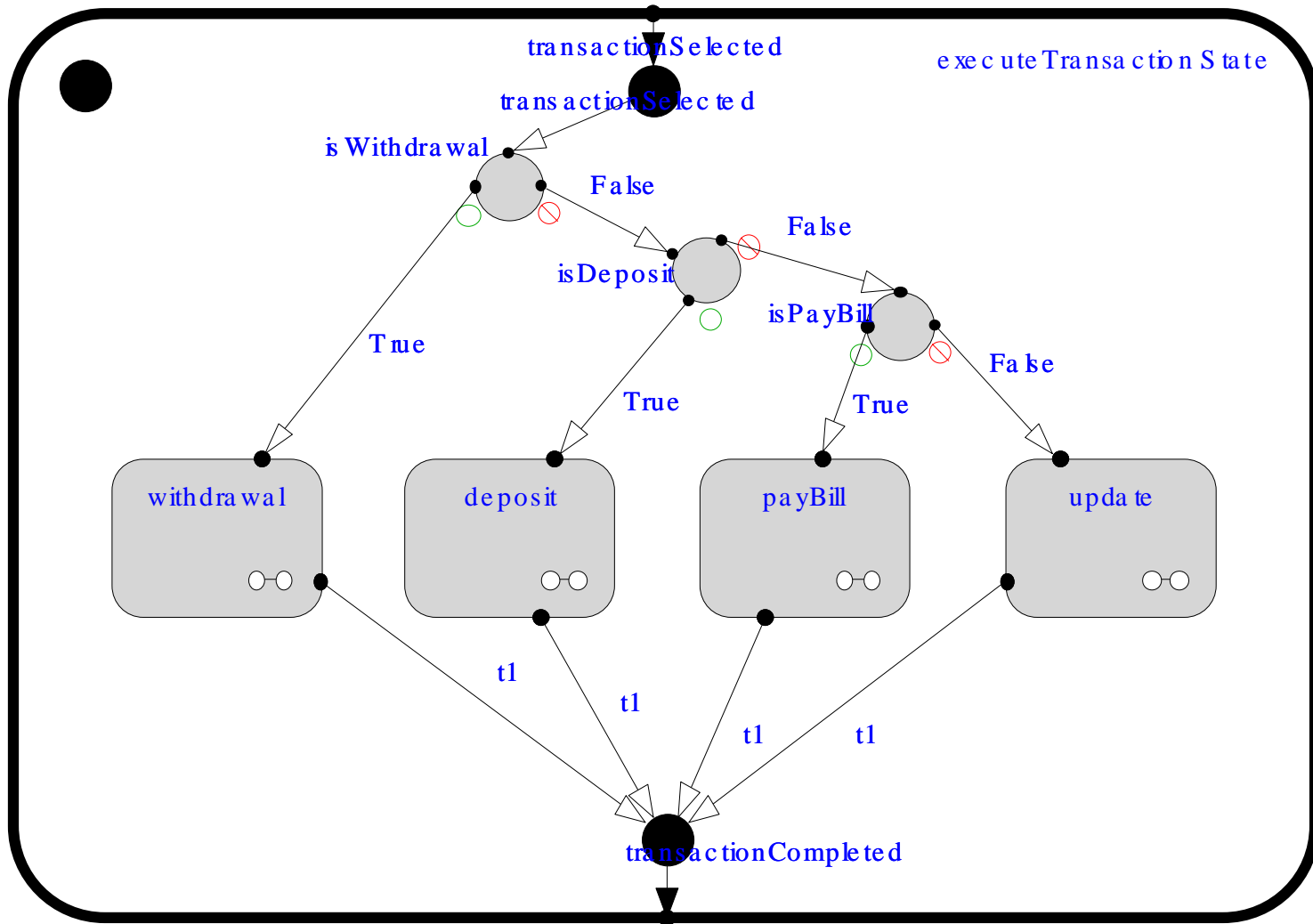
Examples include:

- **Does not exist** in Use Cases
- **Not supported** in Message Sequence Charts
- **Not supported** in UML interaction diagram
- **Not supported** in UML activity diagrams
- **Basic path interaction notation** in Use Case Maps

Temporal Dependency (Follows)

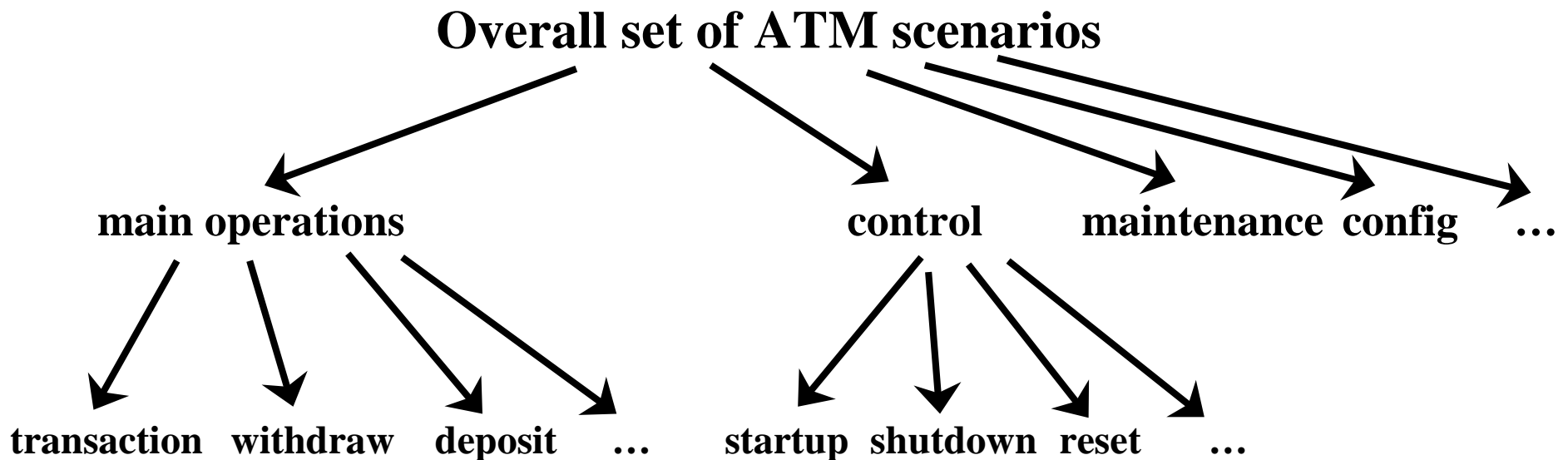


Temporal Dependency (Mutual Exclusion)



Functional Dependency

Capture the coexistence of two or more scenarios inside a same conceptual (or logical) cluster



Examples include:

- **Can be described by means of scenario (or use case) packages in UML, MSC, and UCM**

Related Approaches

- **Synthesis algorithms perform automatic generation of state machines from a set of interaction diagrams**
- **Bruce Douglass' state machine patterns**

Summary

- **We identified four different types of inter-scenario relationships**
- **Our approach is based on the use of patterns**
- **It proceeds from inter-scenario relationships to hierarchical state machine design**
- **It allows considering non-functional requirements**

Future Work

- **Define a catalogue of patterns (or pattern language) for hierarchical state machine design that deal with the different types of relationships**
- **Formalize temporal relationships**
- **Get more implementations of the patterns in industrial and academic contexts**