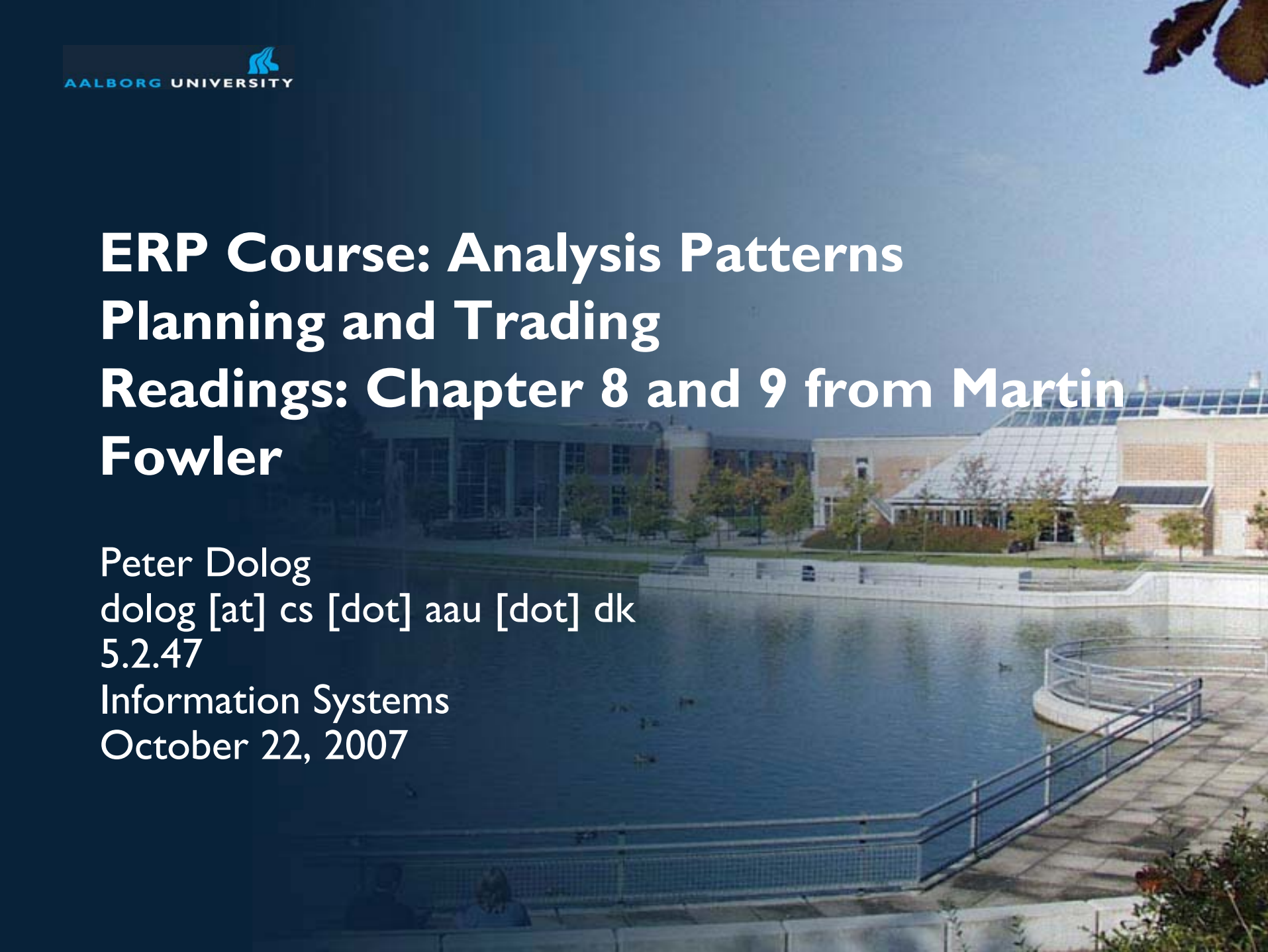# ERP Course: Analysis Patterns Planning and Trading
Readings: Chapter 8 and 9 from Martin Fowler

Peter Dolog
dolog [at] cs [dot] aau [dot] dk
5.2.47
Information Systems
October 22, 2007

Planning

Plan is a set of actions which will be carried out by resources
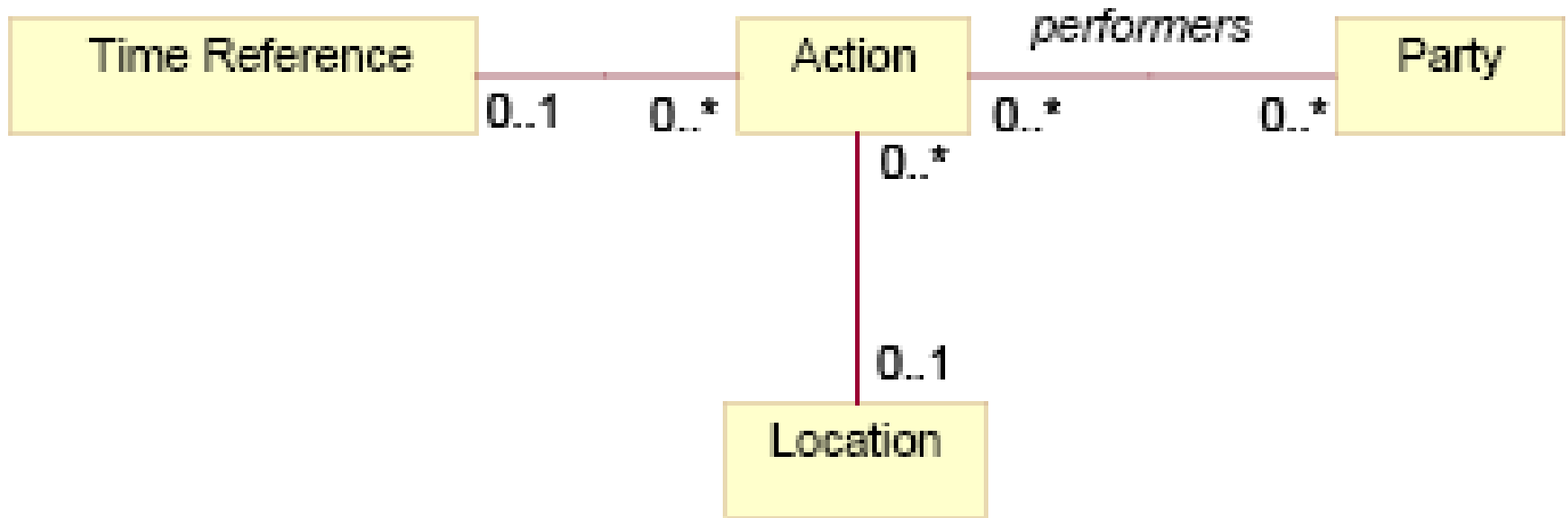Actions are planed, completed, abondoned,
    postponed/suspended
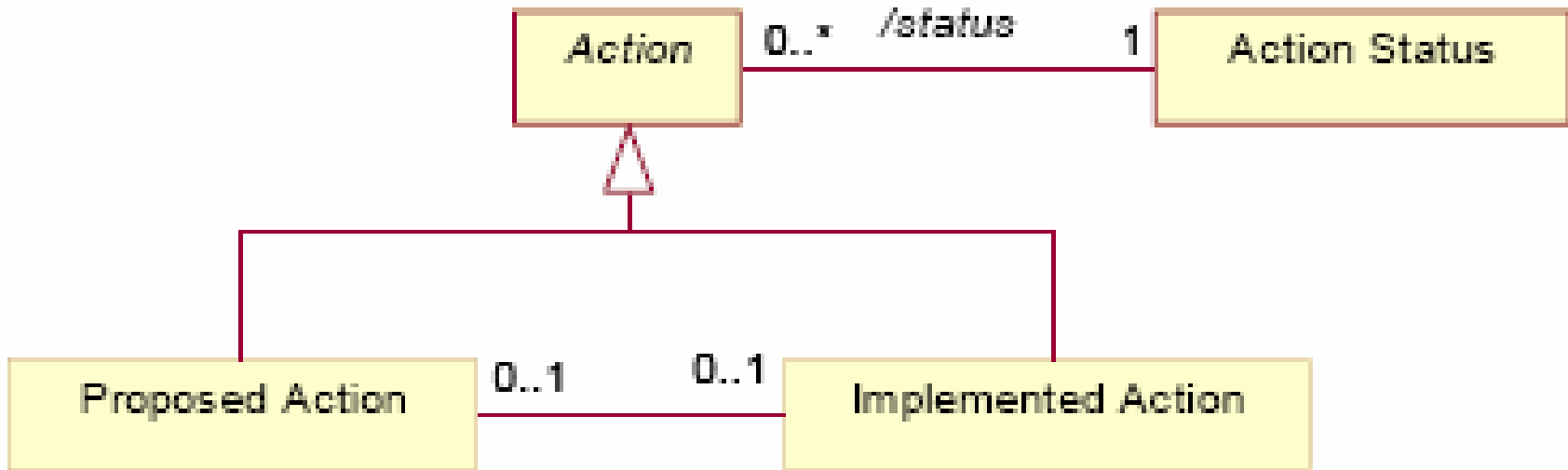They have resources allocated
A protocol is a set of actions repeated many times
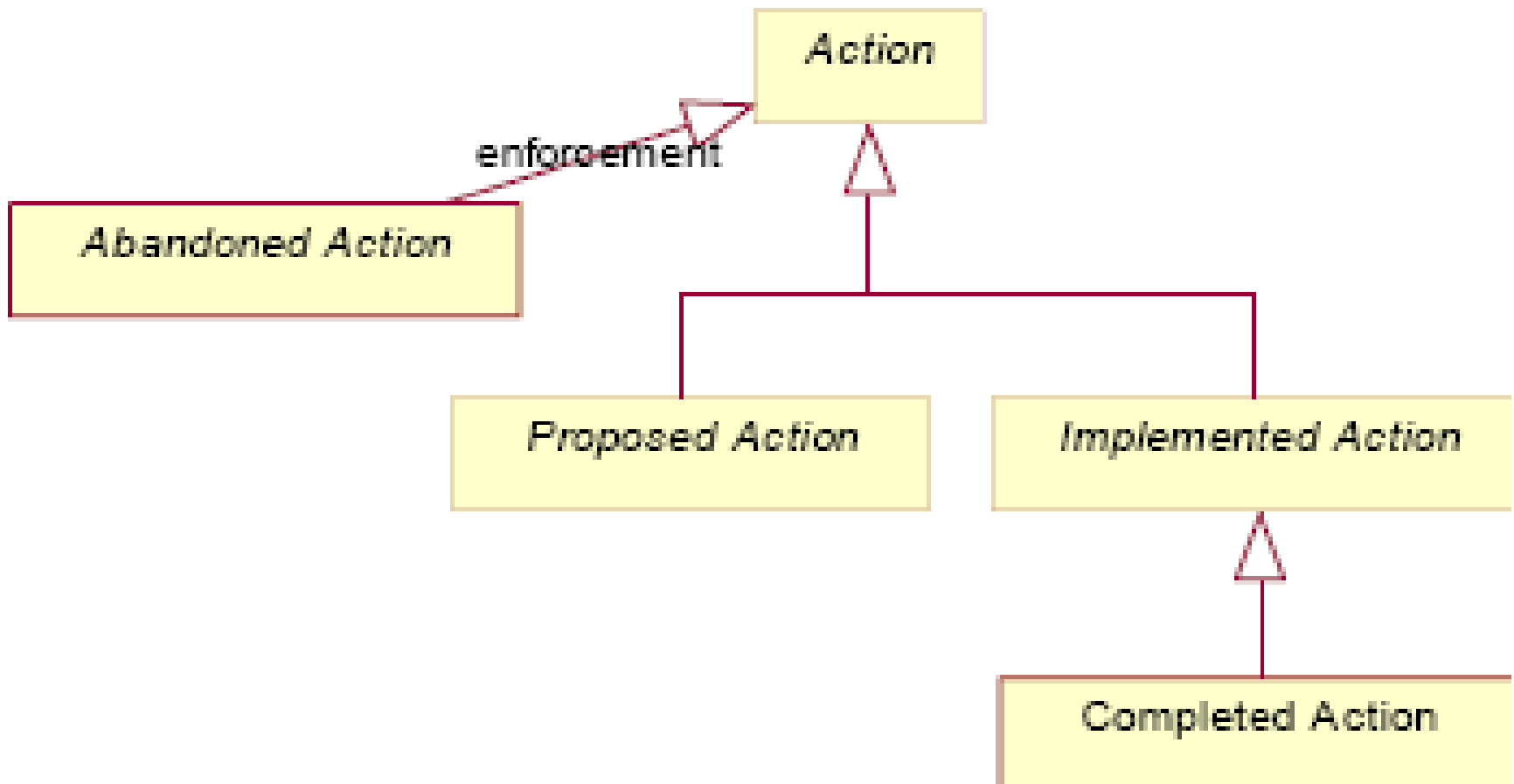Activities have outcomes and conditions to start

Properties of Actions

Basic Structure of Plans and Actions

Complemented and Abandoned Actions

Suspension of Actions
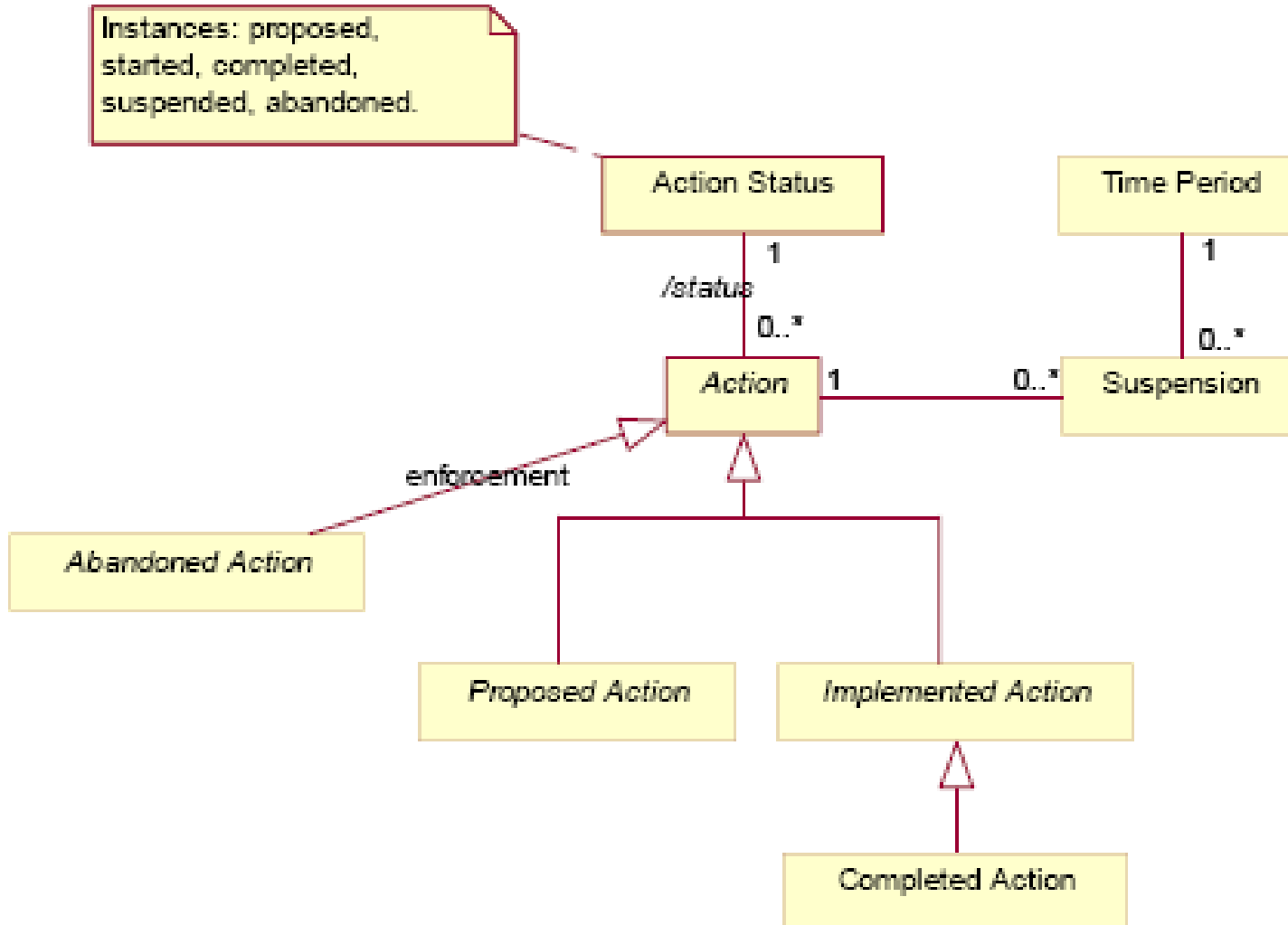
A Plan Consisting of References to Proposed Actions
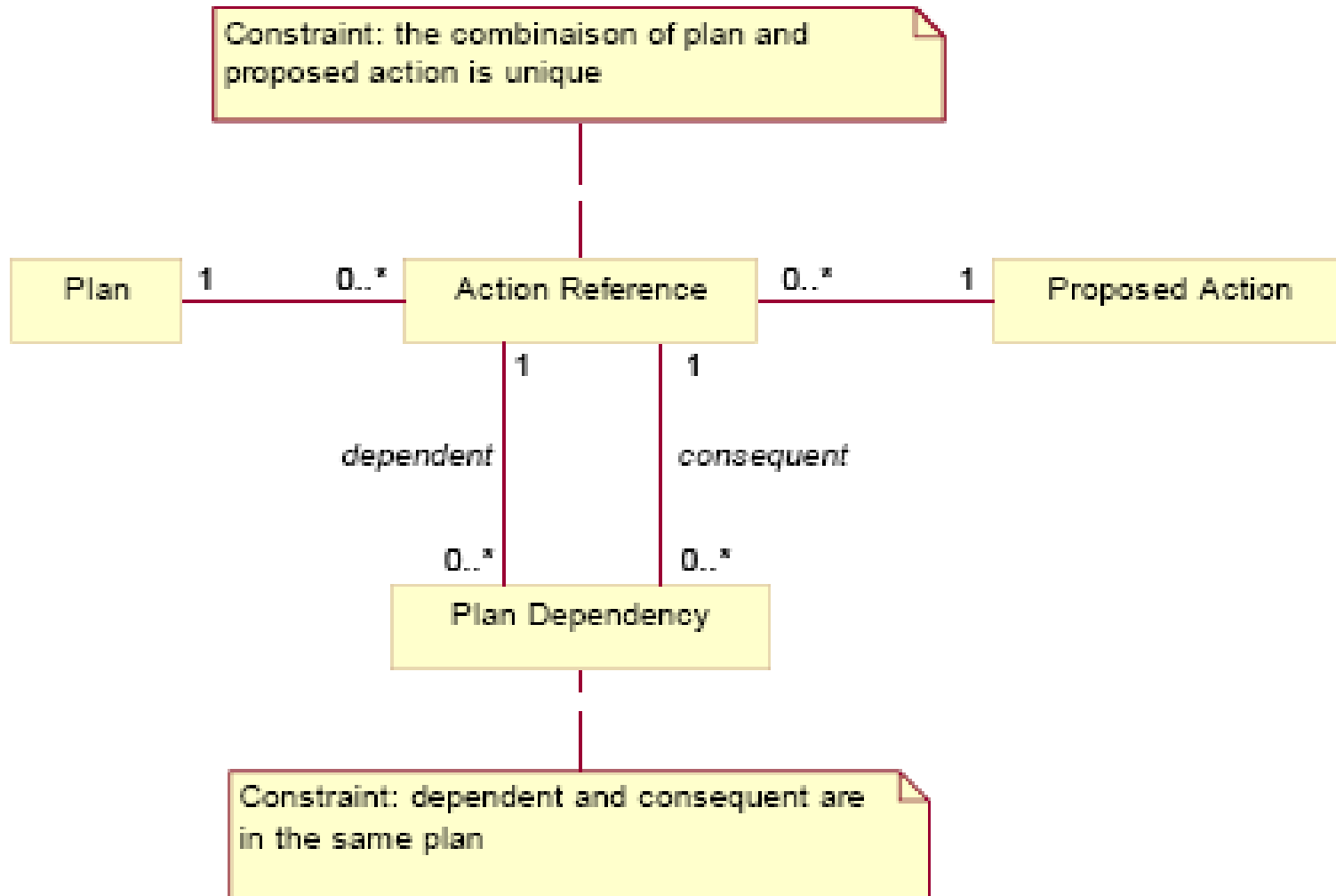
Replacement Plans

replacements

0..*			
Plan	1 0..*	Action Reference	0..* 1 Proposed Action
0..*			
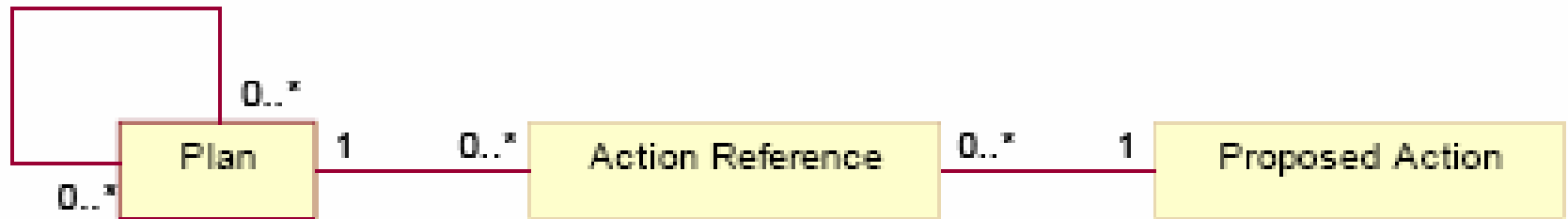
Plans as Actions and Compound Actions
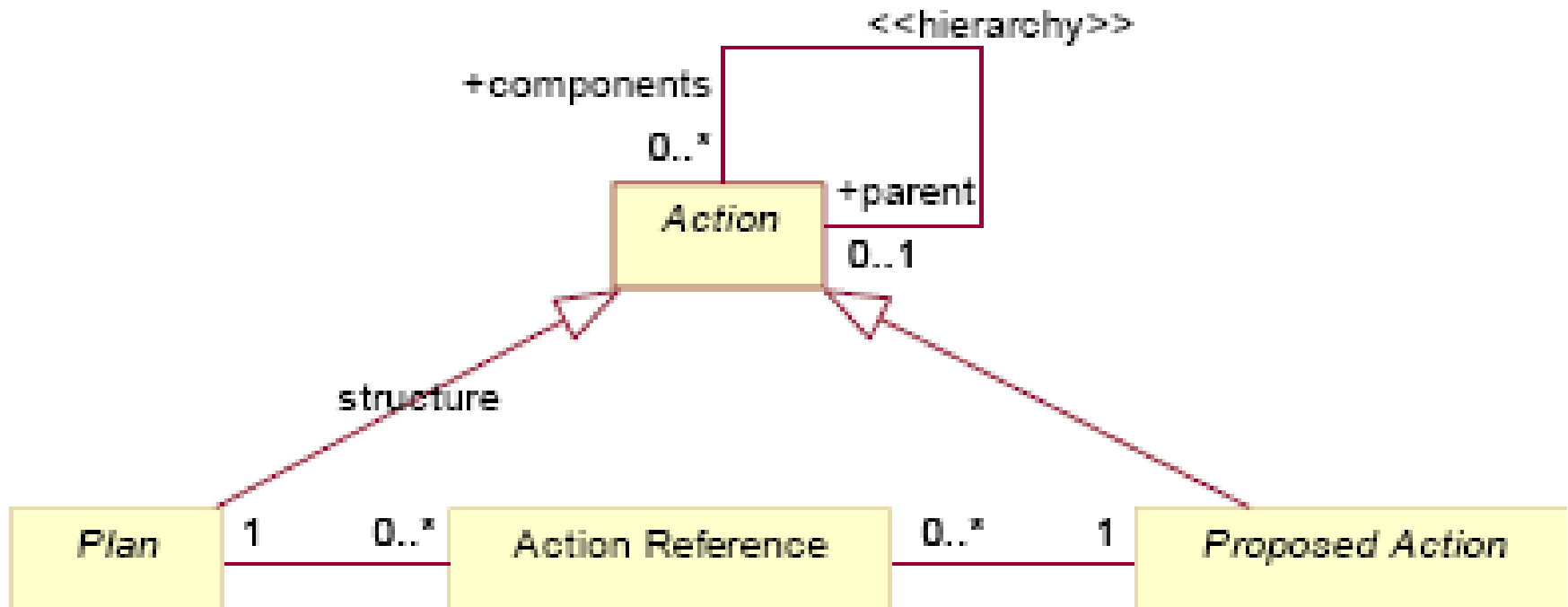
Structure for Protocols

Relationships among Action, Plan, and Protocol
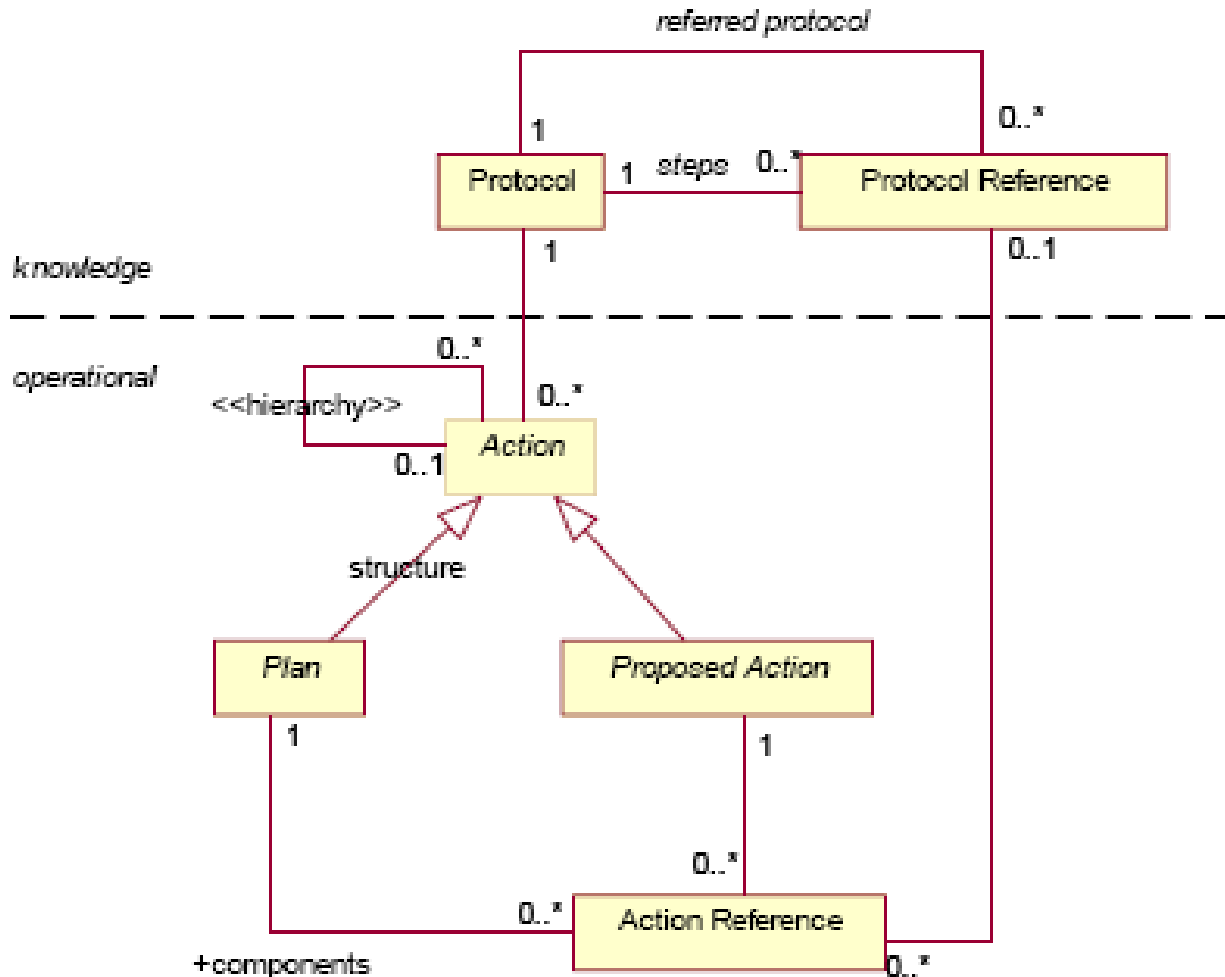
Plan as a **DAG** of Proposed Actions
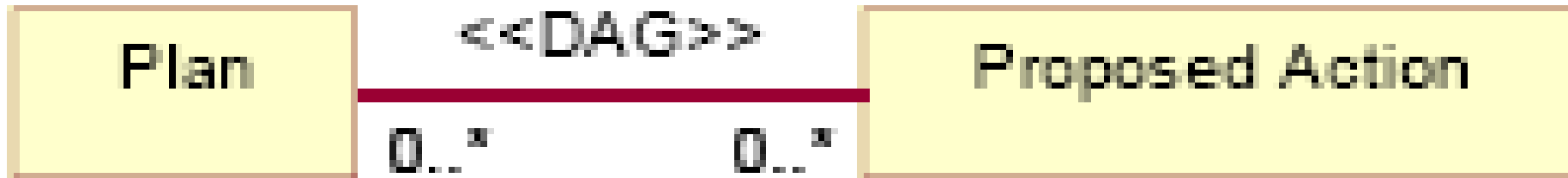
Protocol as DAG
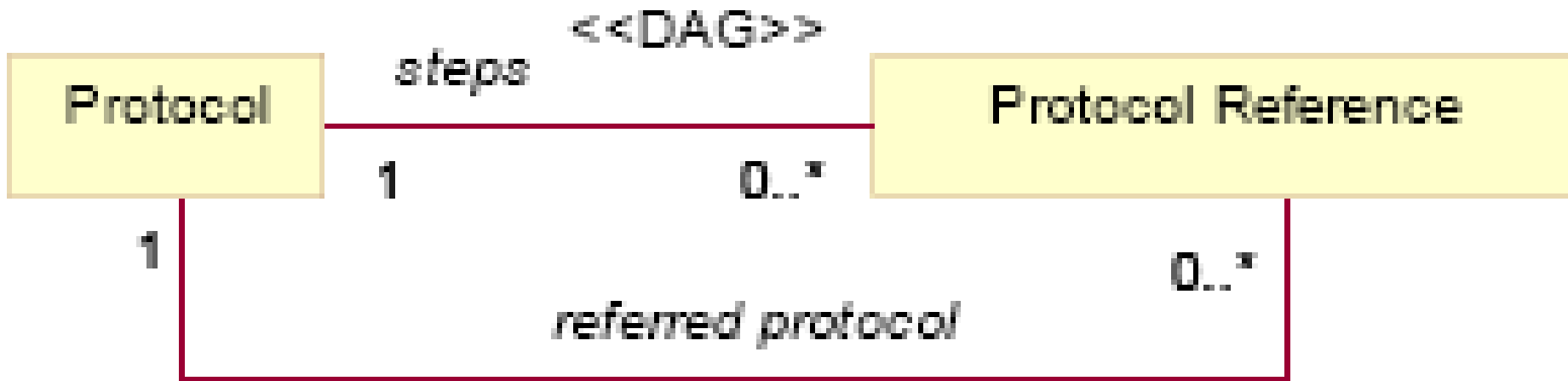
Action's Use of Resources
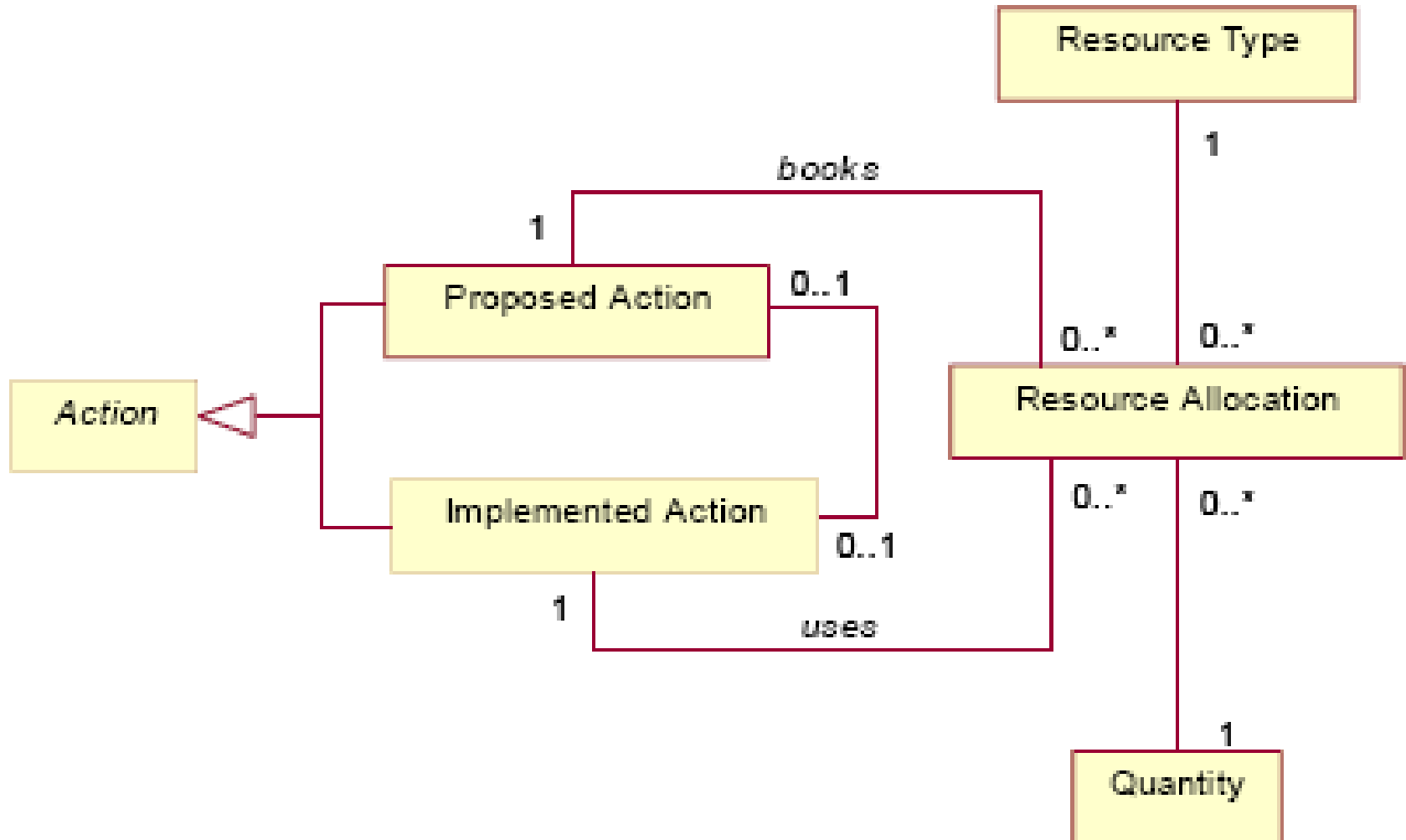
Resource Allocation for Assets
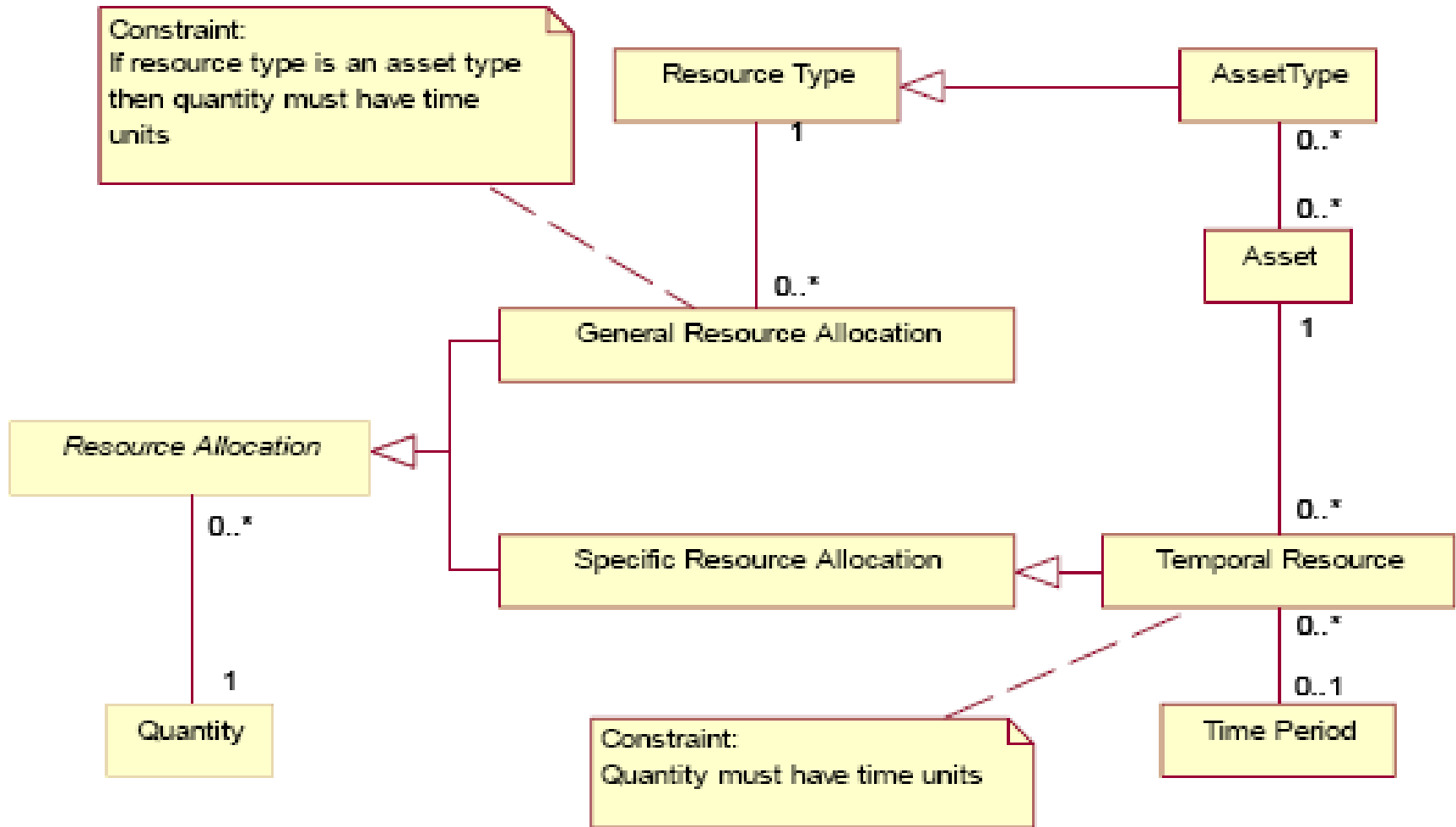
Allowing Specific Allocations of Consumables
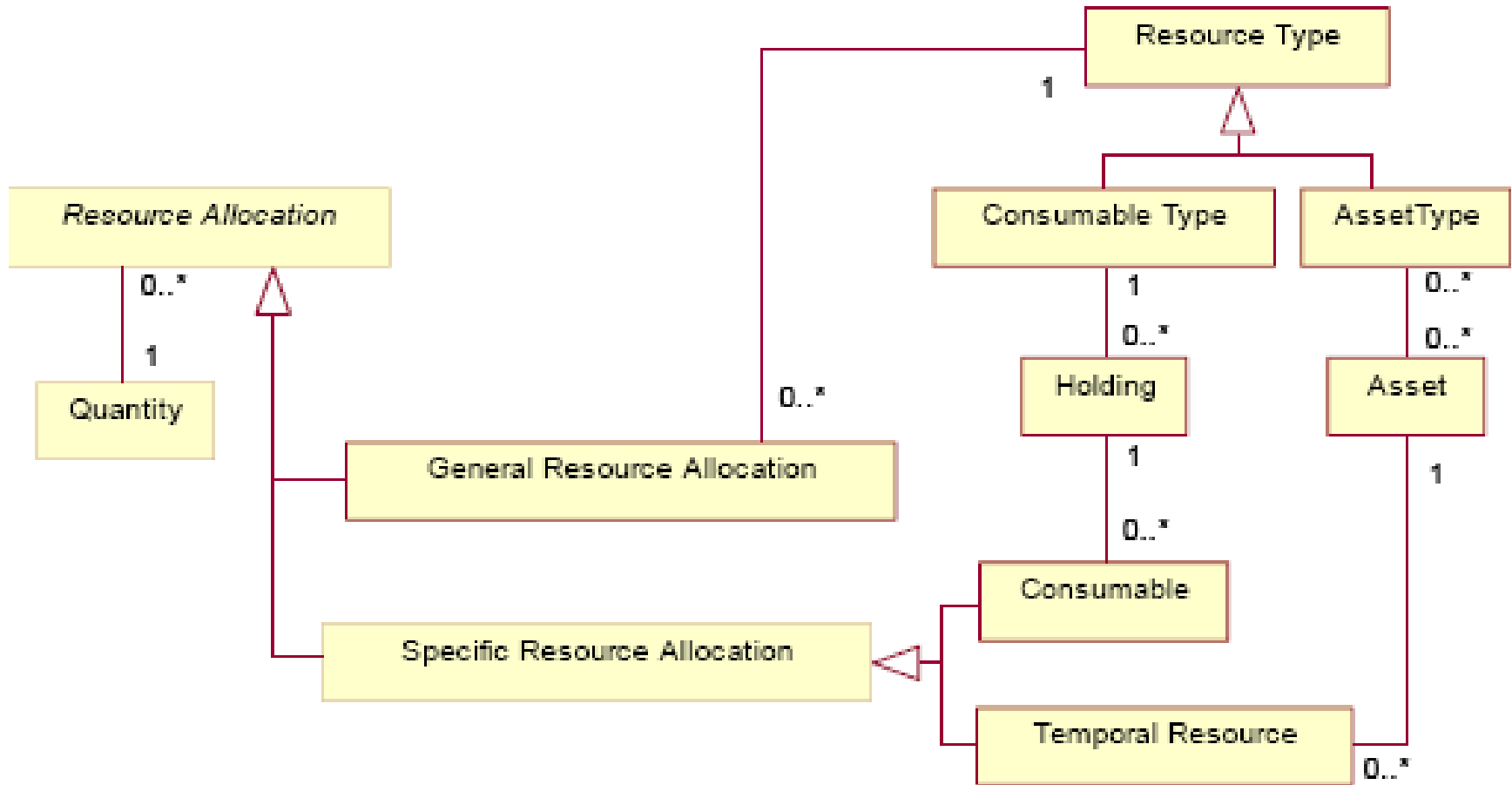
Links Between Observation, Plan, and Action
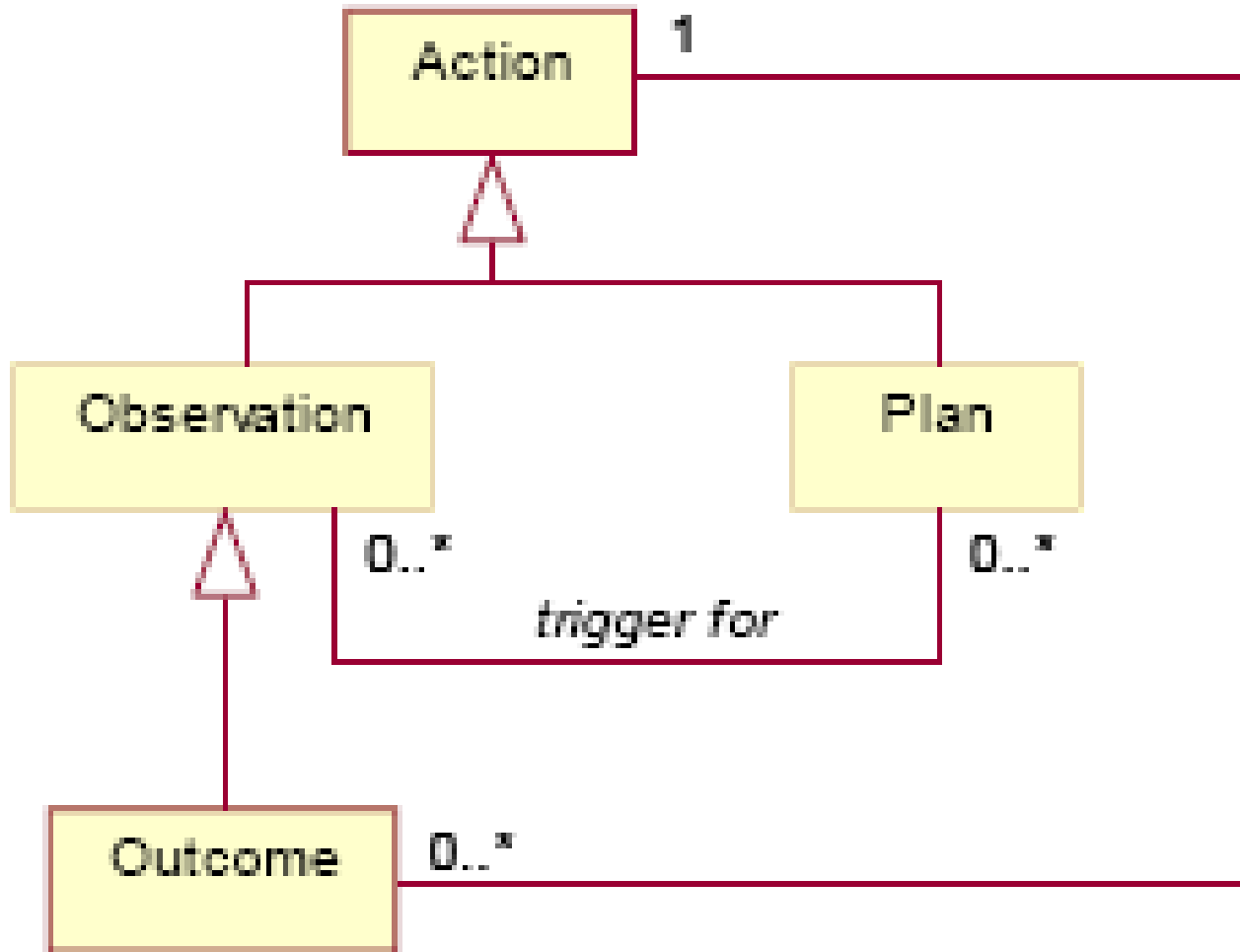
The Use of Start and Outcome Functions at the Knowledge Level
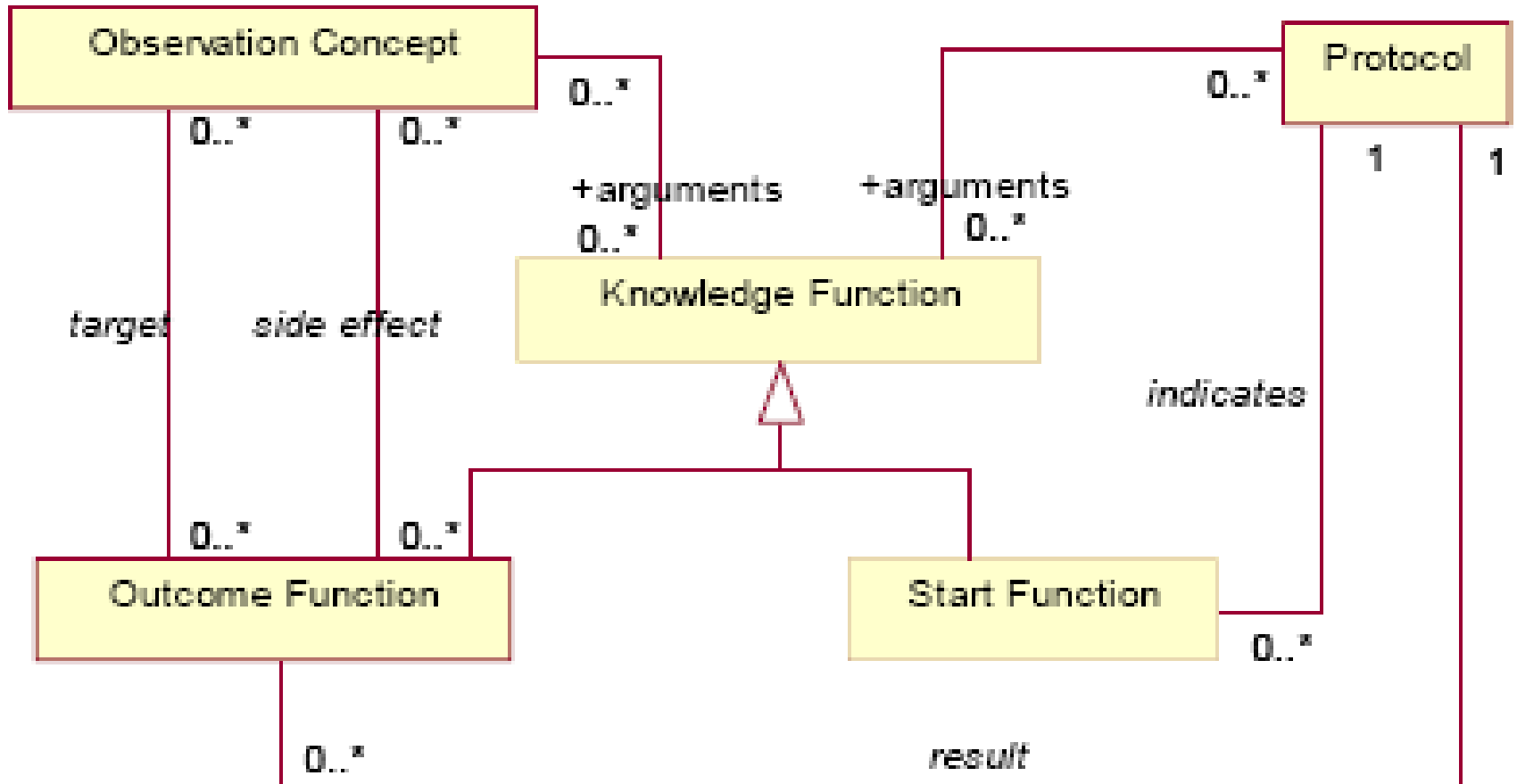
Trading

Trade is about buying and selling

Trade is described by a contract

Contracts can be grouped into portfolios

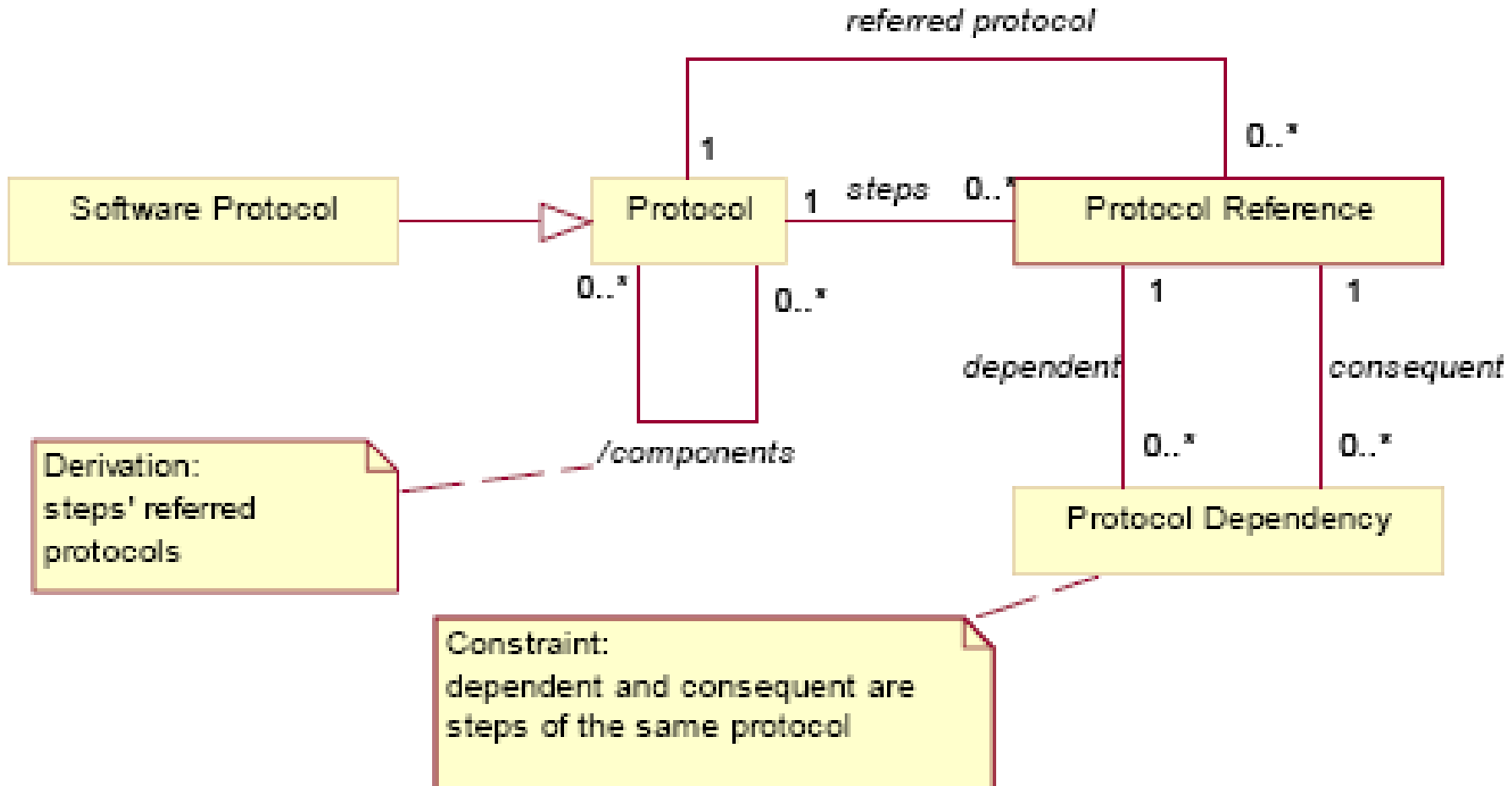Prices of goods change according to whether you look at them from selling and buying point of view – quotes

Rapid changes can be captured by scenarios

Contract

```
Party  +counterparty    Contract          Instrument
                      +amount : Number
                      +price : Money
    1..1        *                      *      1..1

                           △

              Long              Short
```
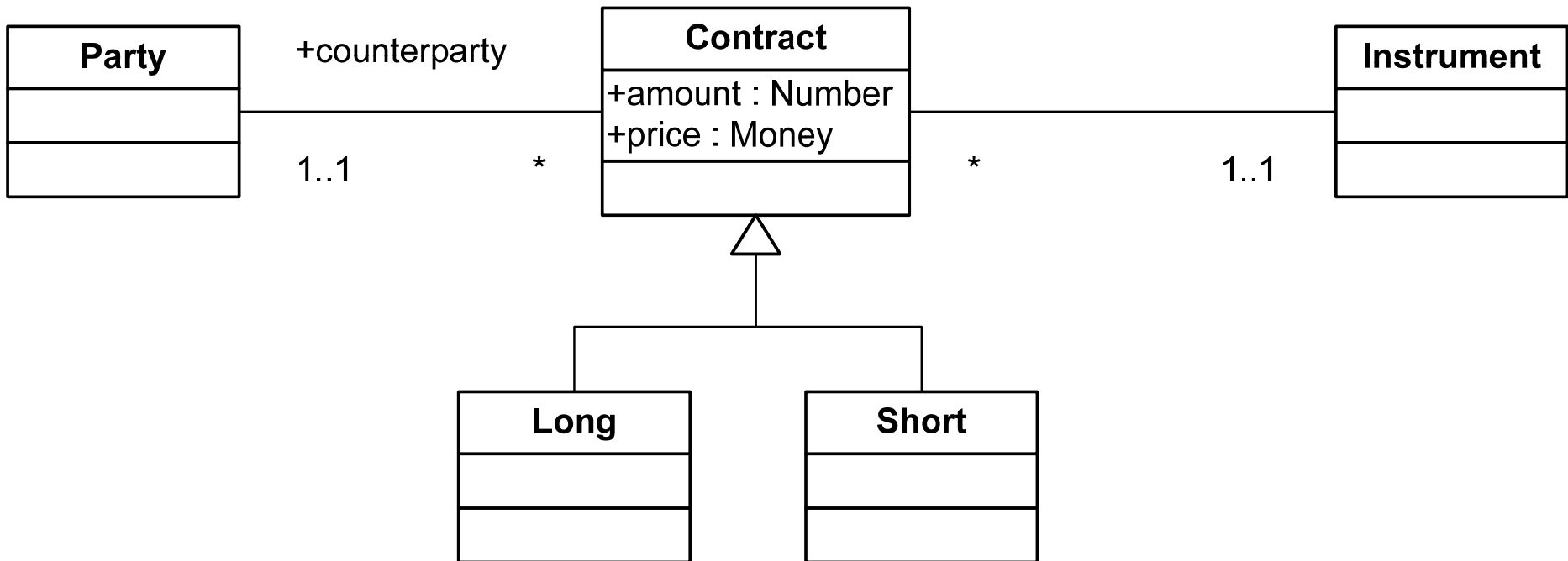
Buyers and Sellers and Relationship
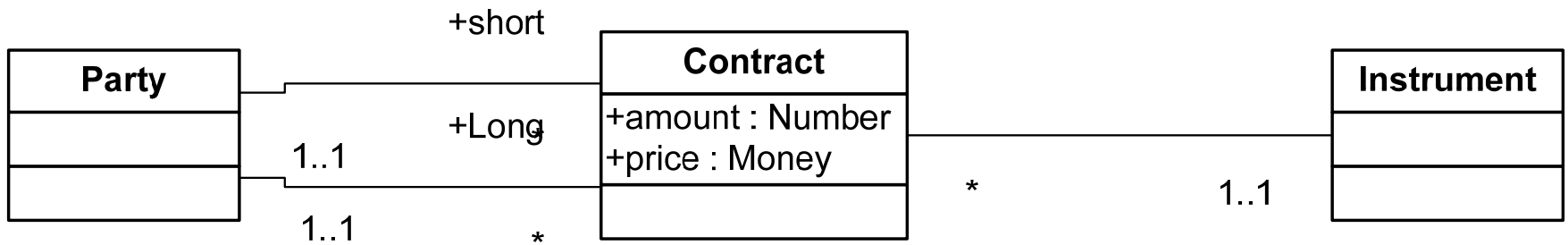
Counterparty and Primaryparty
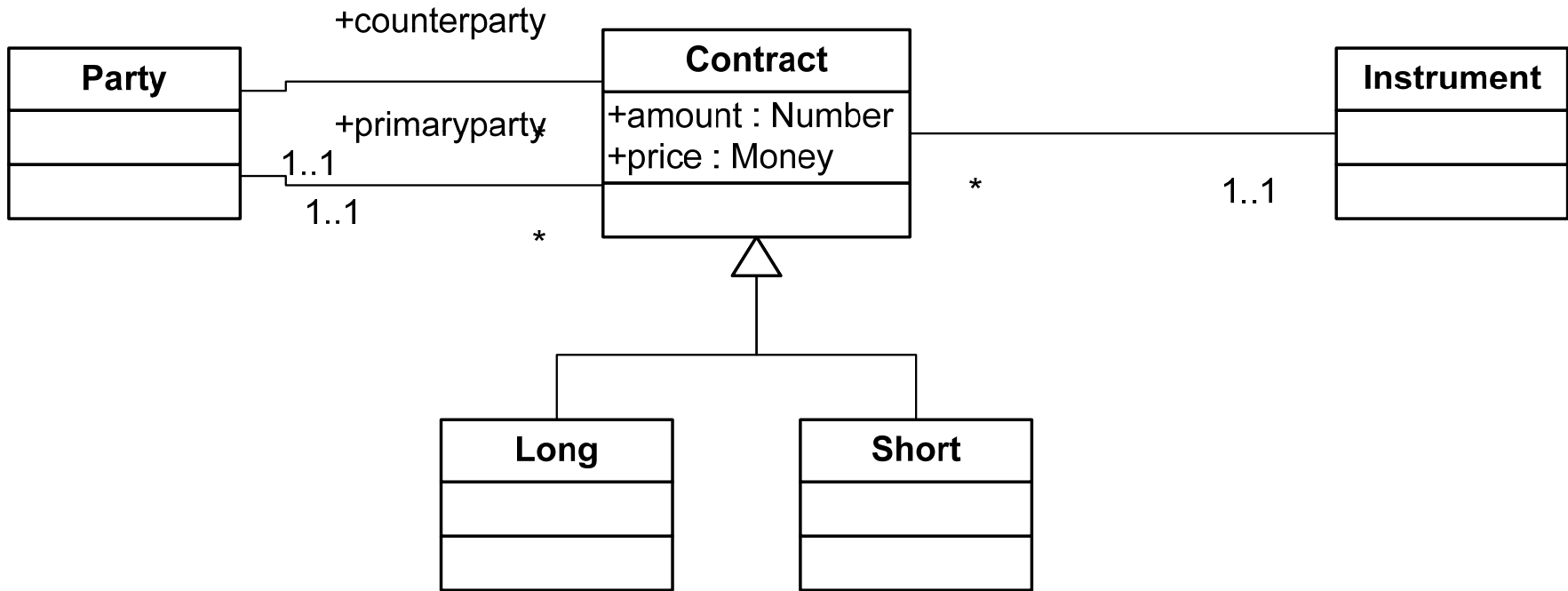
Party mappings

Portfolio

Portfolio
-Value(scenario)

?
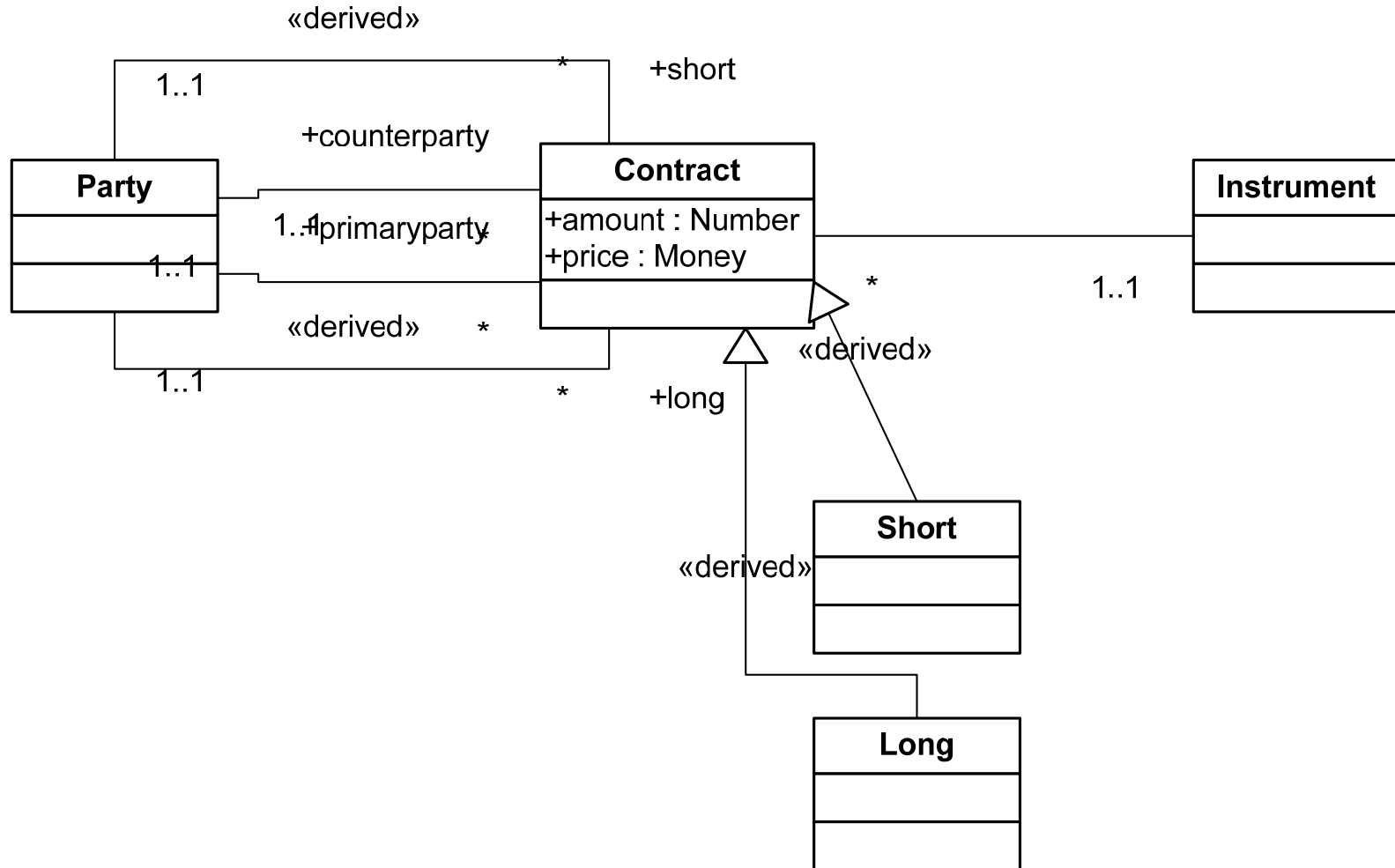
Contract
-Value(scenario)

*
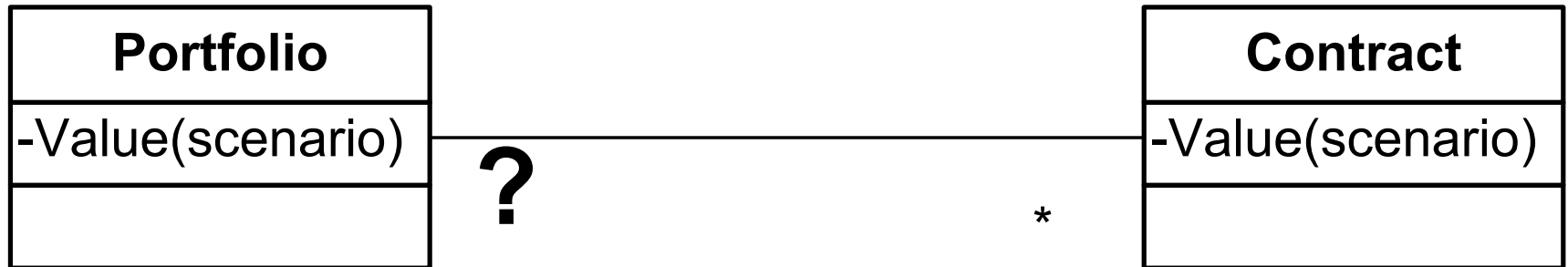
Dynamic portfolios with filters

Portfolio
-Value(scenario)

«derived»

Contract
-Value(scenario)

*   *

{All instances on contract on which self.filter is true}

1..1   +filter

BooleanMethod

Contract Selectors

Date

Instrument   ContractSelector   Party

0..1   0..1   +start   +end

+primaryparties

+counterparties
*   *

*   *   *   *

1..1   +filter

*

Portfolio   «derived»   Contract
-Value(scenario)   -Value(scenario)

*   *
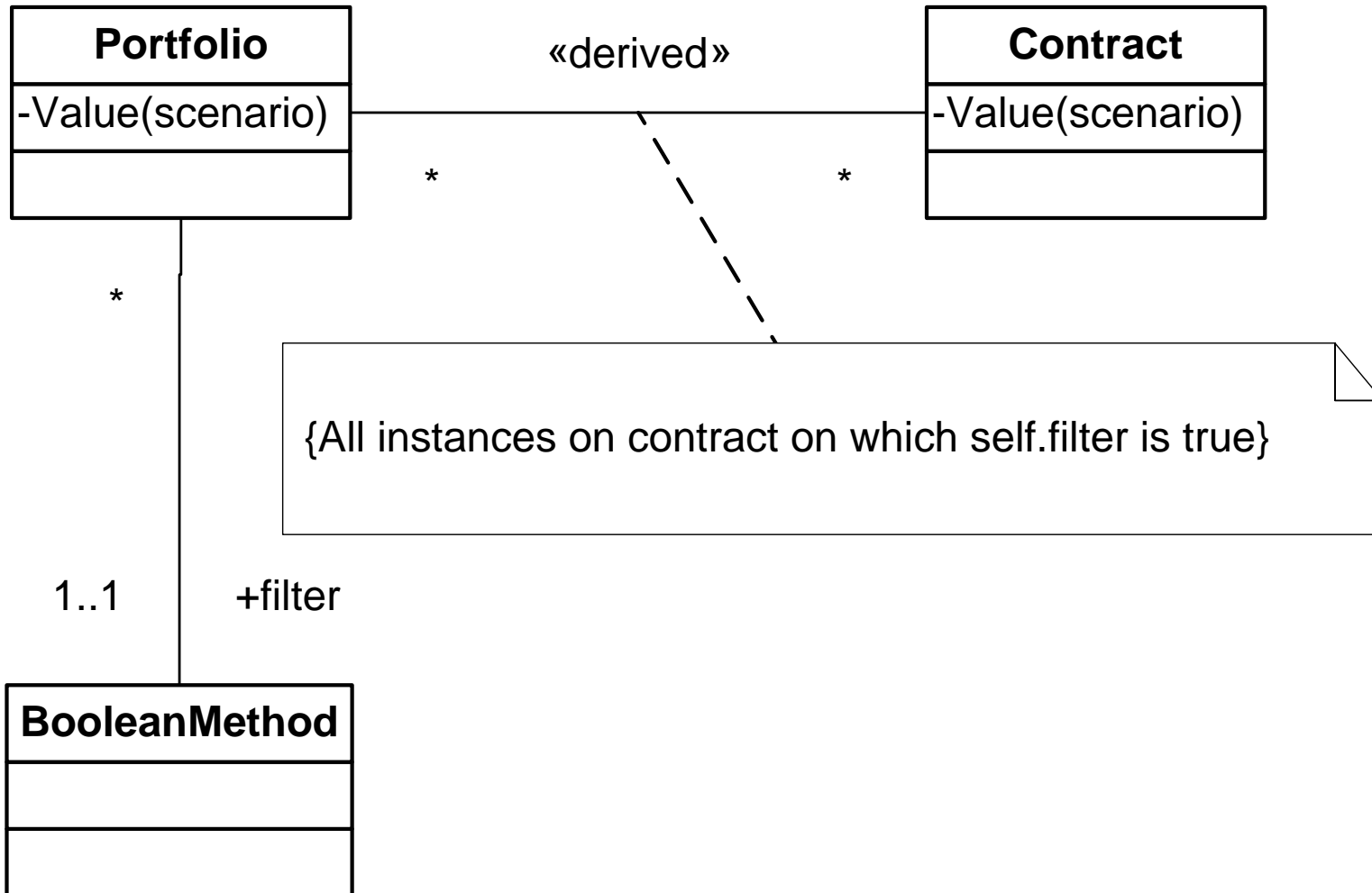
{All instances on contract on which self.filter is true}

Several filter types

Date

```
           0..1      0..1  +start   +end
```

Instrument   **ContractSelector**   +primaryparties   **Party**
                                                             +counterparties
                                          *   *   *
                                 *   *
                                          *   *

HardCodedFilter   *ContractFilter*
-Value(scenario)   -isIncluded
                                       +selectContracts(in Collection)

                                          1..1

                                           *

          *

                                       Portfolio   «derived»   **Contract**
1..1   +filter   -Value(scenario)   -Value(scenario)
                                                          *   *

BooleanMethod
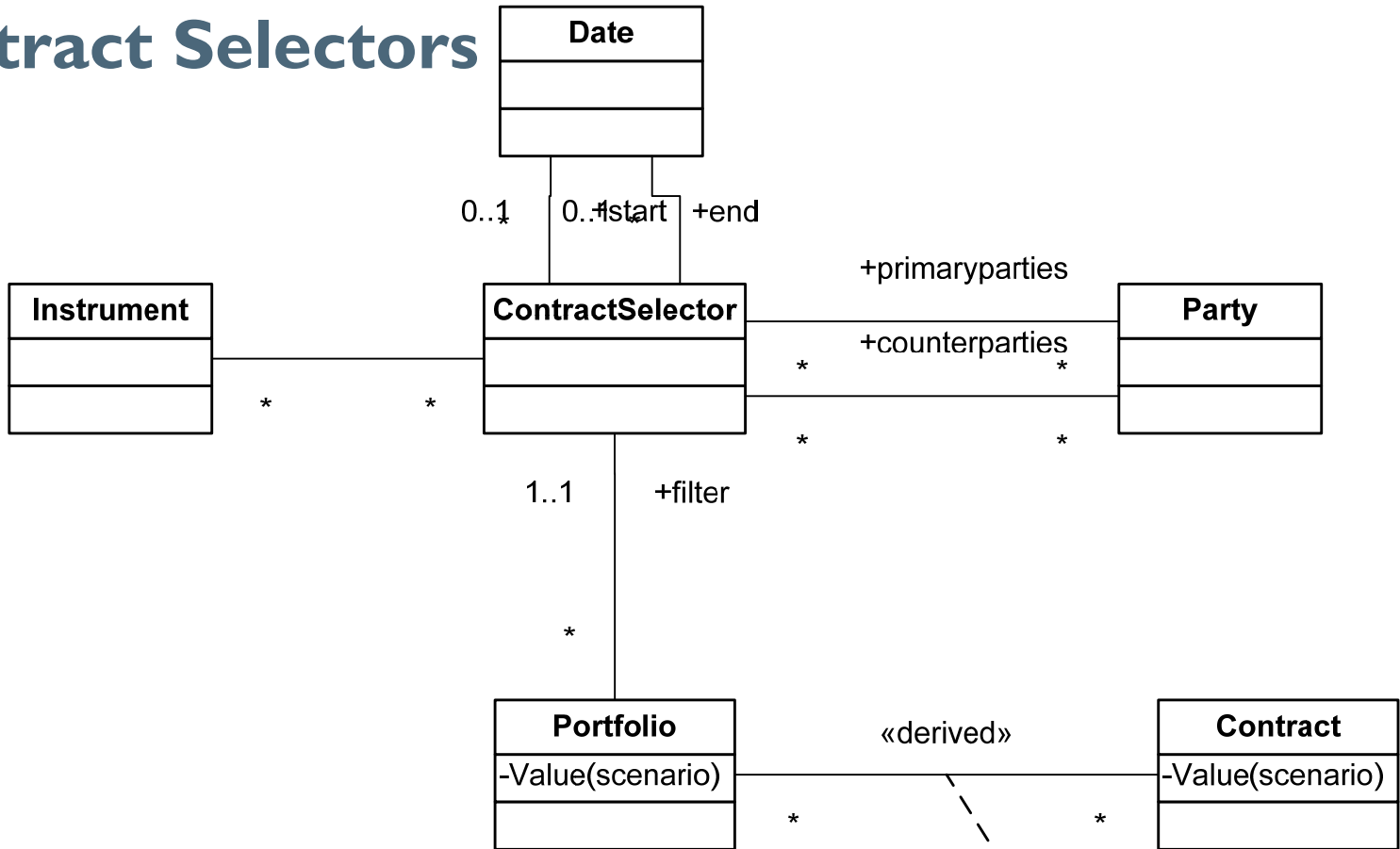
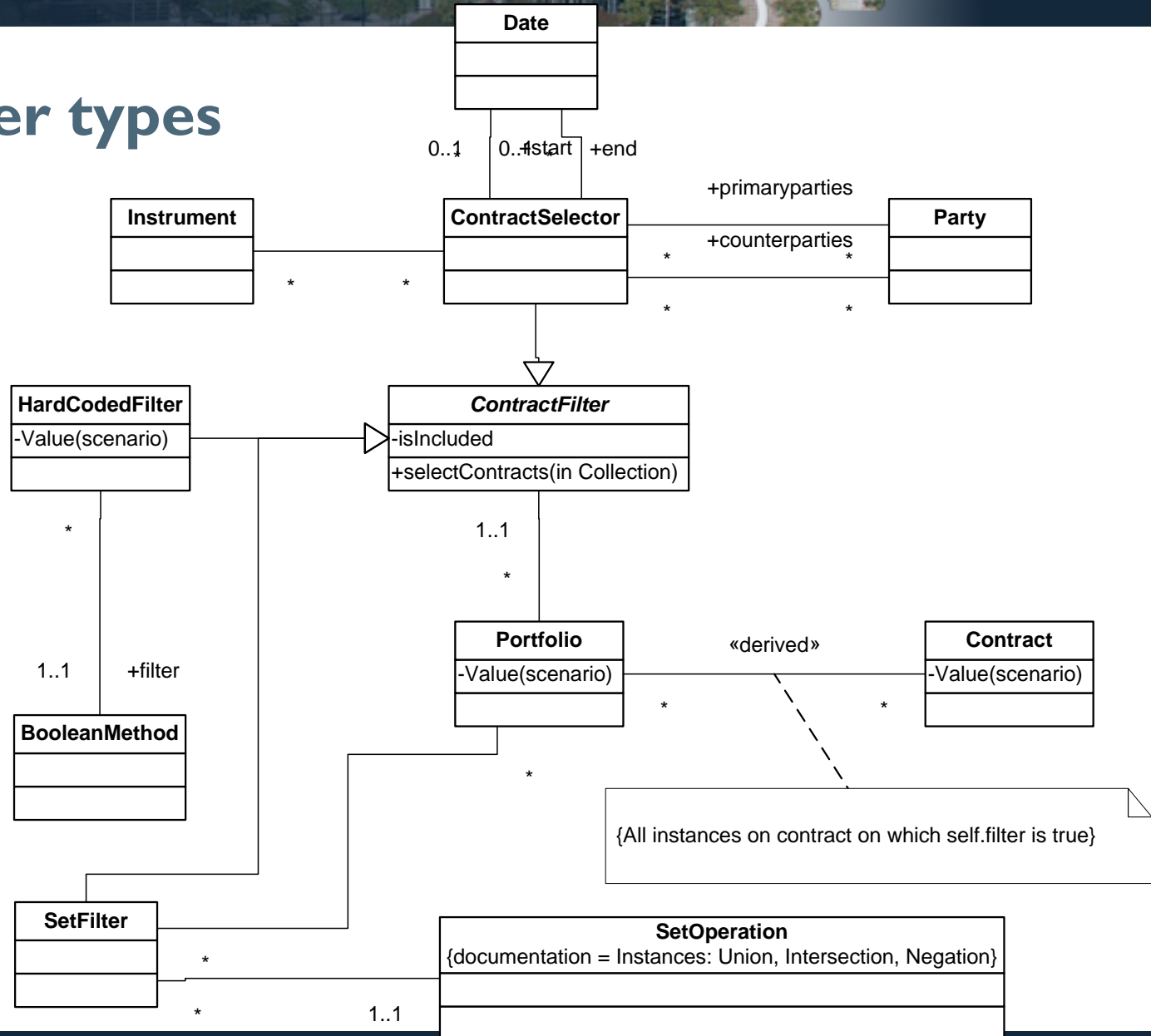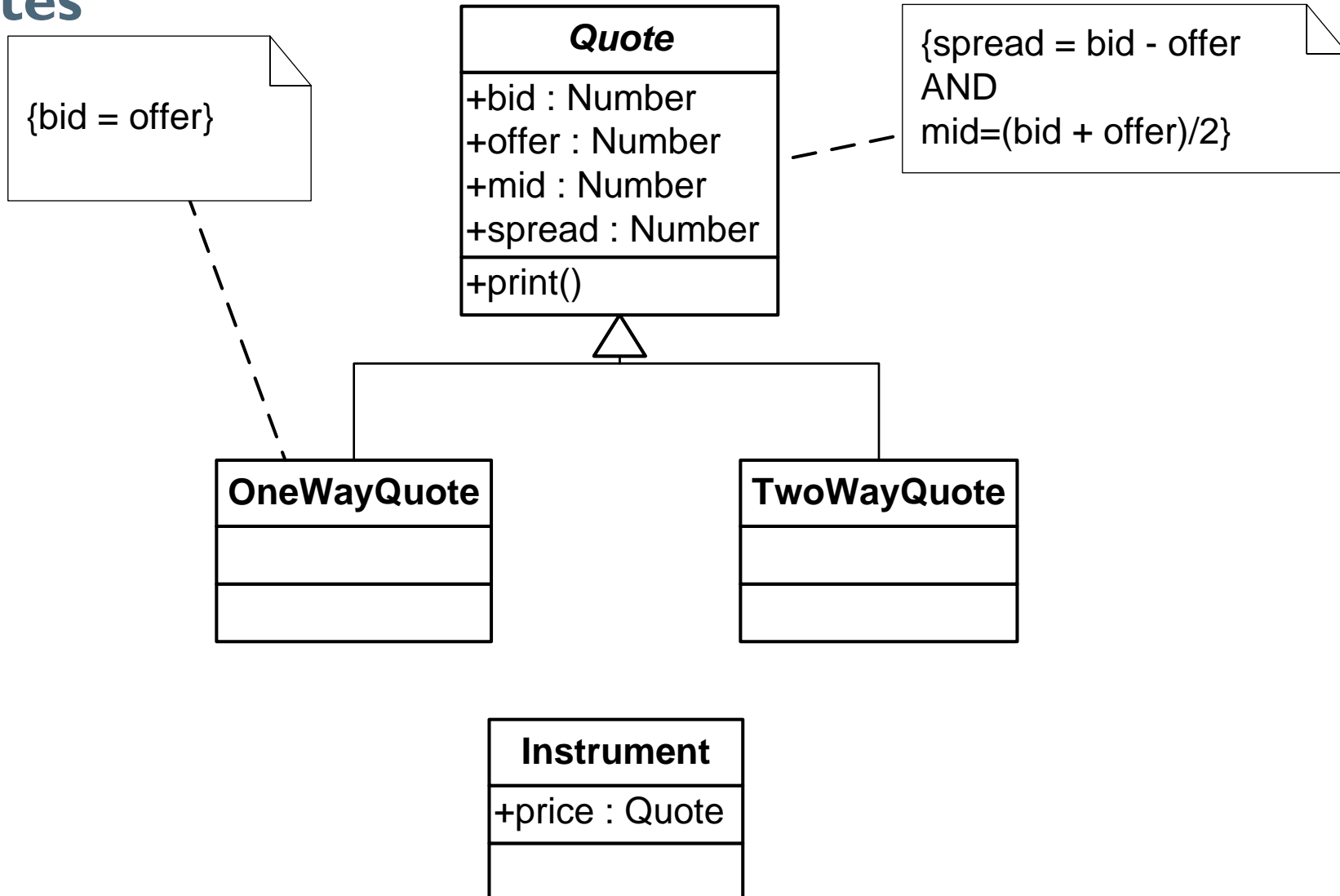   *

                                                              {All instances on contract on which self.filter is true}
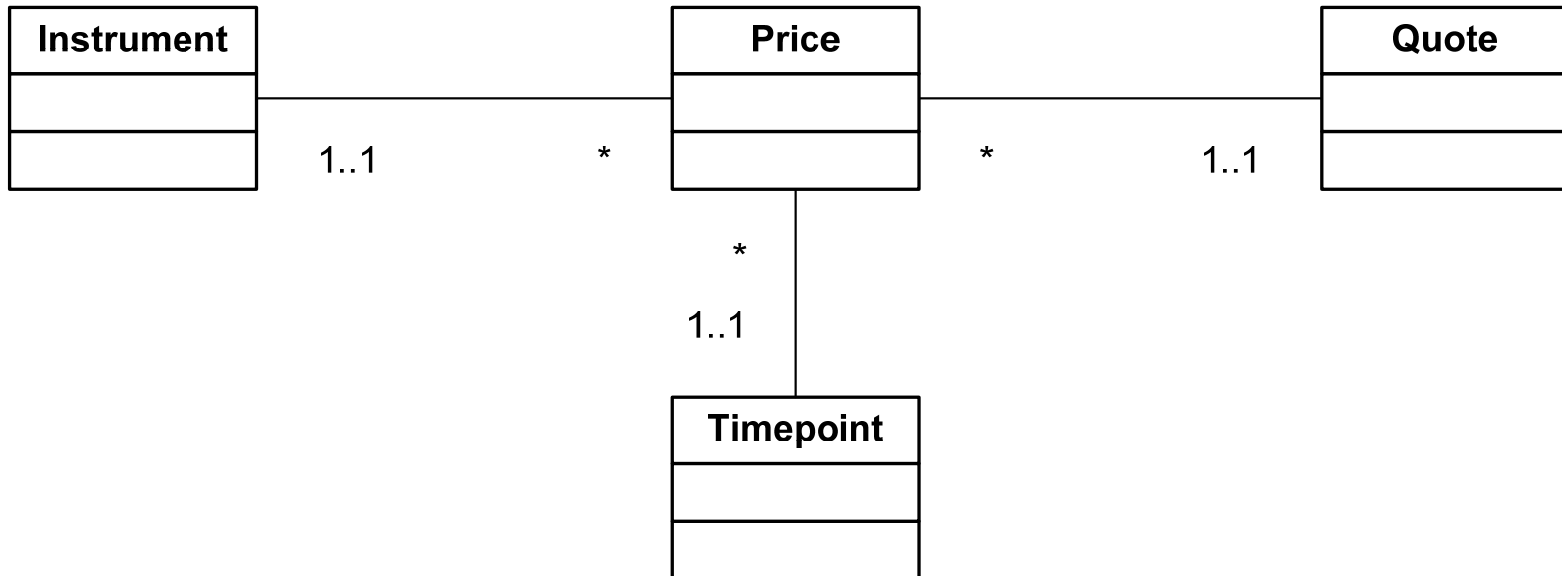
SetFilter   **SetOperation**
                                       {documentation = Instances: Union, Intersection, Negation}
              *

         *   1..1

Quotes

{bid = offer}

Quote

+bid : Number
+offer : Number
+mid : Number
+spread : Number

+print()

{spread = bid - offer
AND
mid=(bid + offer)/2}

OneWayQuote

TwoWayQuote

Instrument

+price : Quote

Changes in Quotes

Instrument		Quote		Timepoint

1..1 * * 1..1

Instrument		Price		Quote

1..1 * * 1..1

*

1..1

Timepoint

Scenarios

Derived Scenario Elements