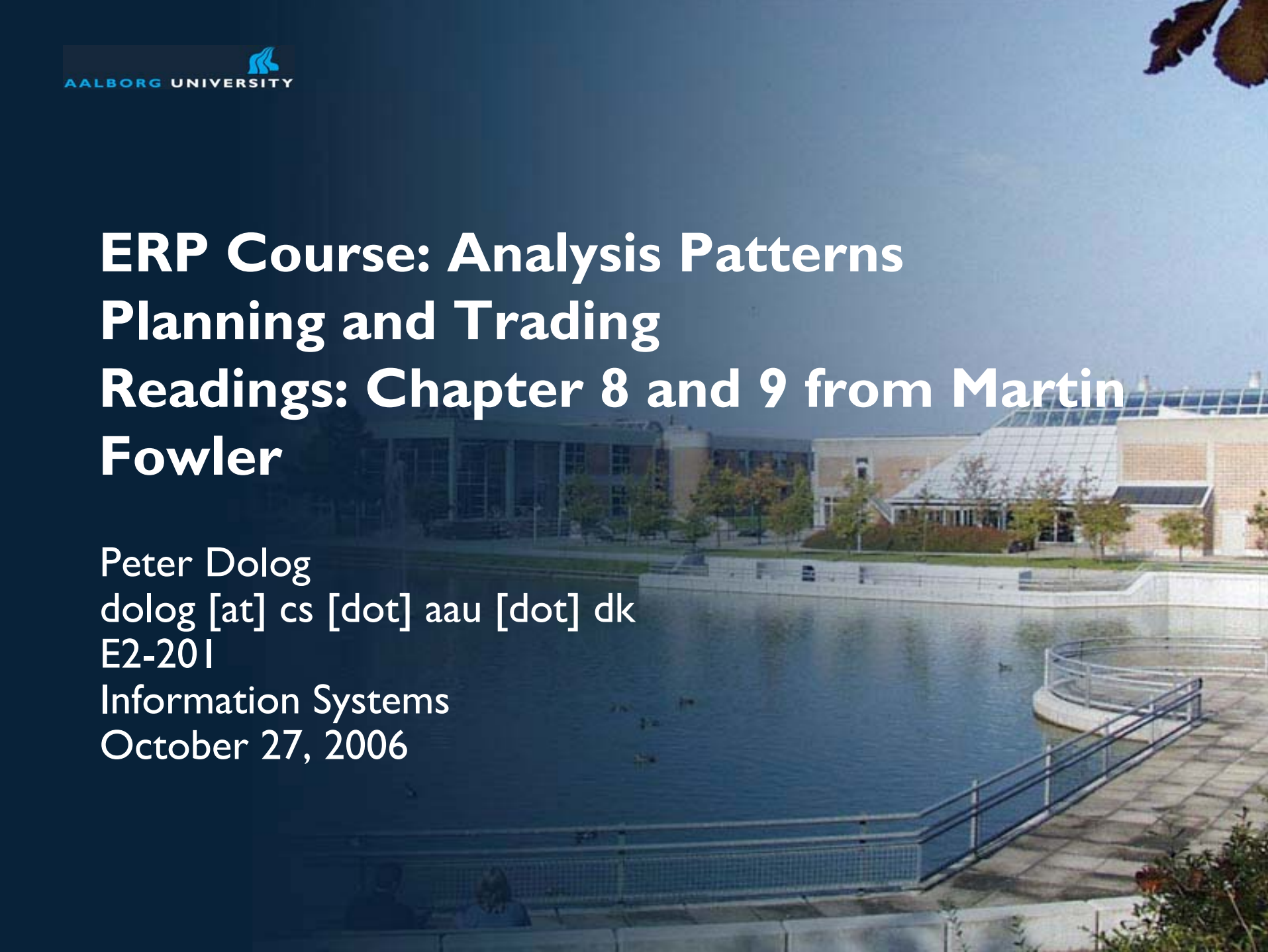# ERP Course: Analysis Patterns Planning and Trading Readings: Chapter 8 and 9 from Martin Fowler

Peter Dolog
dolog [at] cs [dot] aau [dot] dk
E2-201
Information Systems
October 27, 2006

Planning

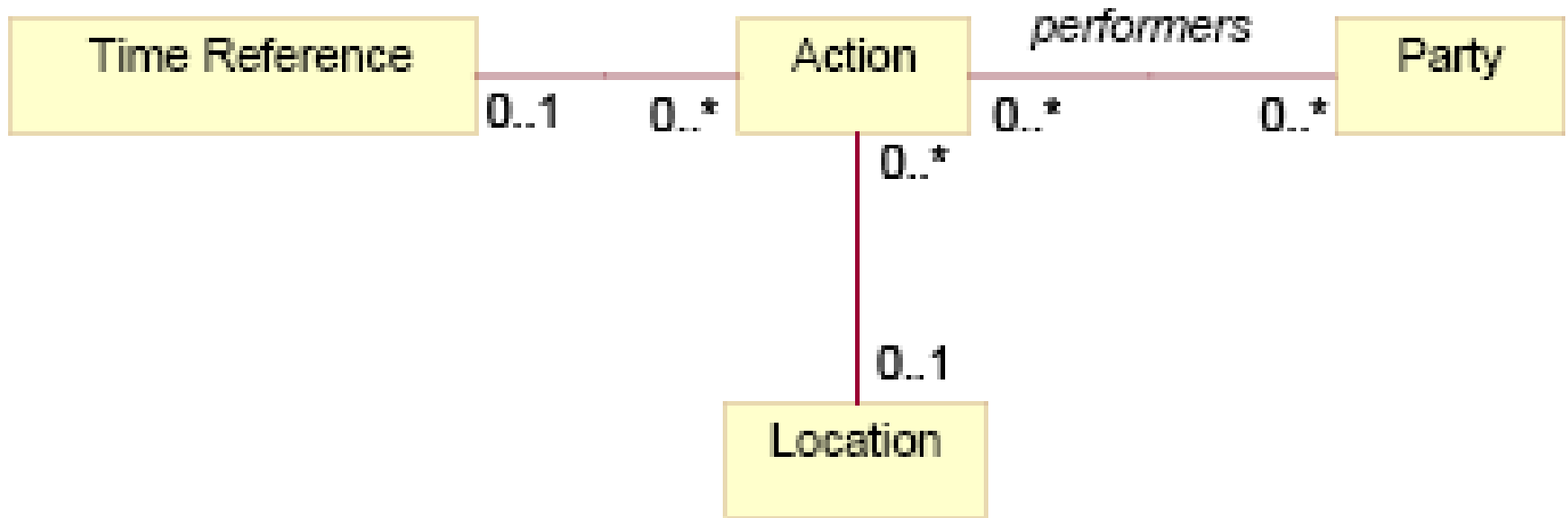Plan is a set of actions which are carried out by resources
Actions are planed, completed, abondoned,
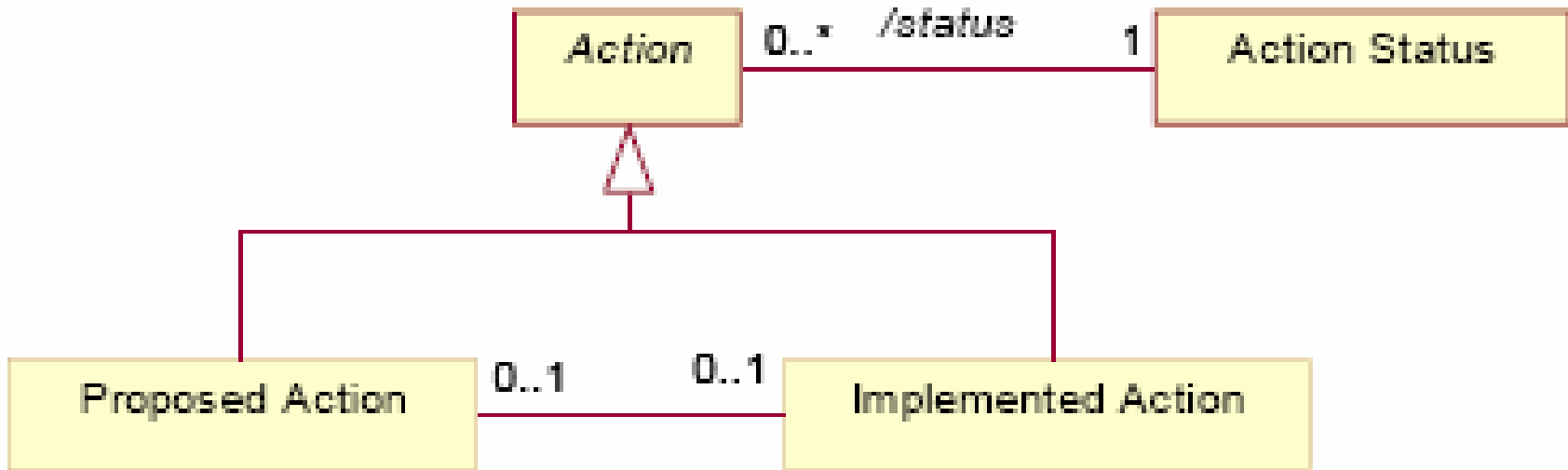   postponed/suspended
They have resources allocated
A protocol is a set of actions repeated many times
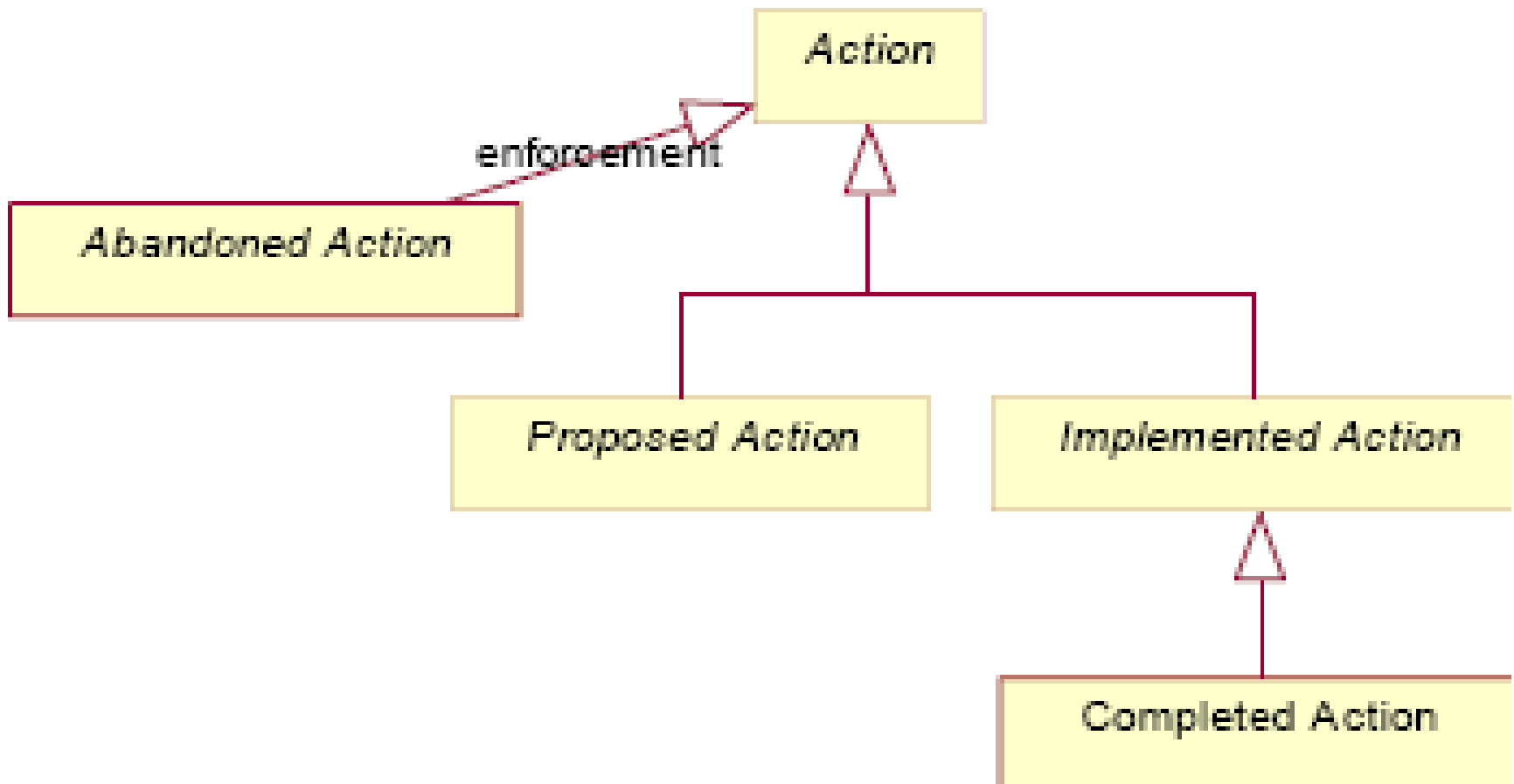Activities has outcomes and conditions to start

Properties of Actions

Basic Structure of Plans and Actions

Complemented and Abandoned Actions

Suspension of Actions
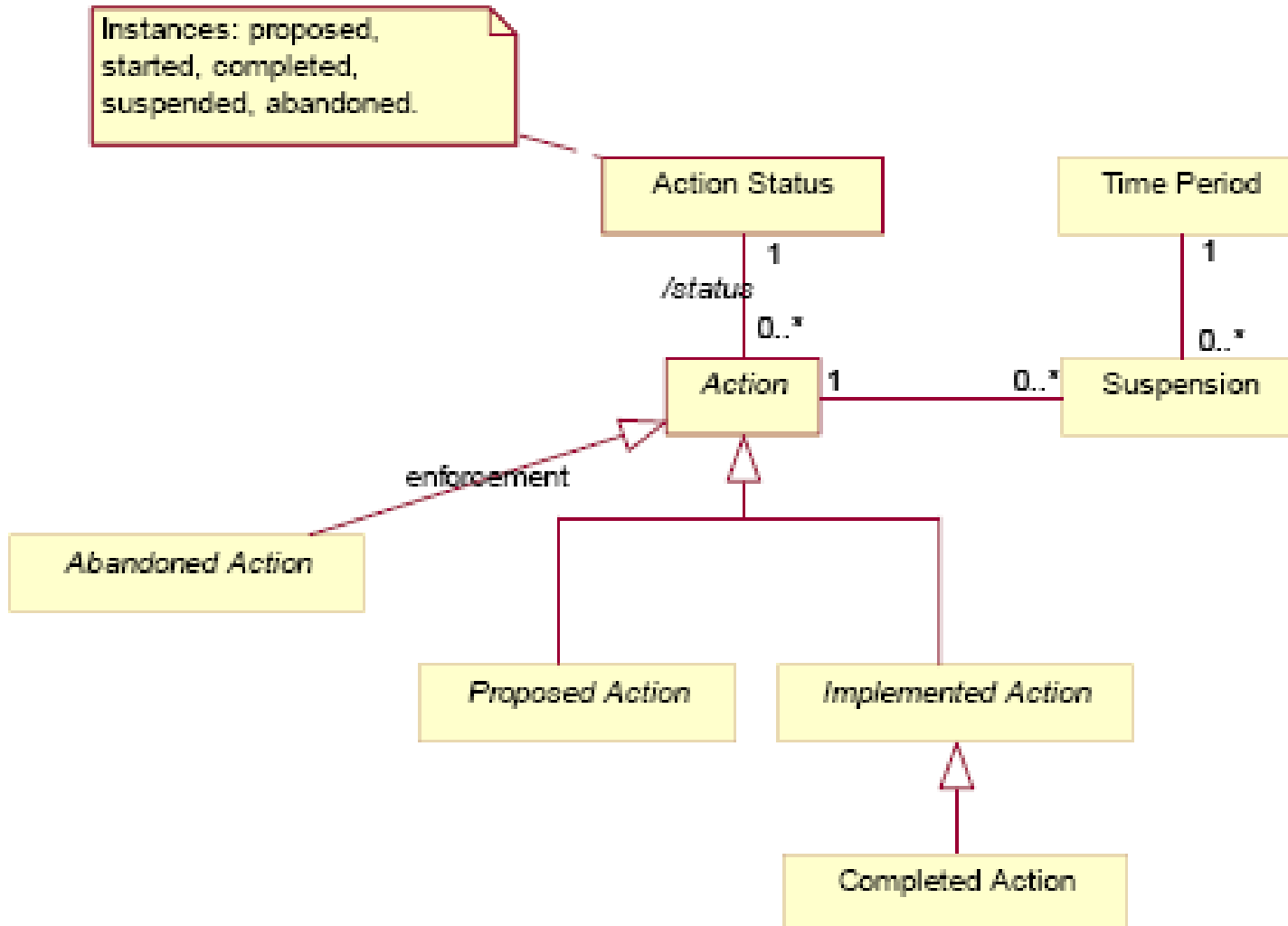
A Plan Consisting of References to Proposed Actions

Constraint: the combinaison of plan and proposed action is unique

Plan — 1 — 0..* — Action Reference — 0..* — 1 — Proposed Action

1 — dependent

1 — consequent

0..* — 0..*

Plan Dependency

Constraint: dependent and consequent are in the same plan
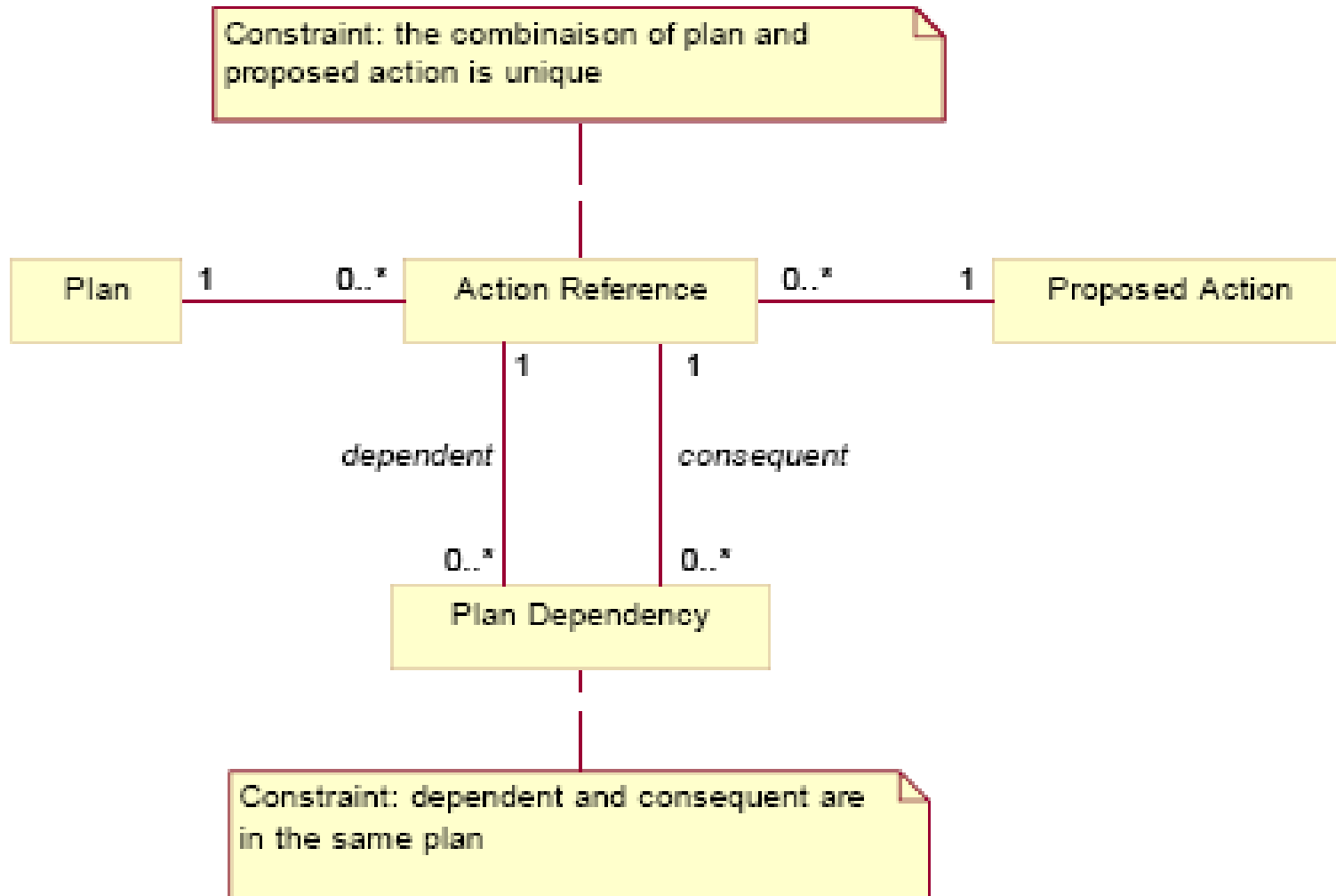
Replacement Plans

replacements

| Plan | 1 ── 0..* | Action Reference | 0..* ── 1 | Proposed Action |

0..*

0..*
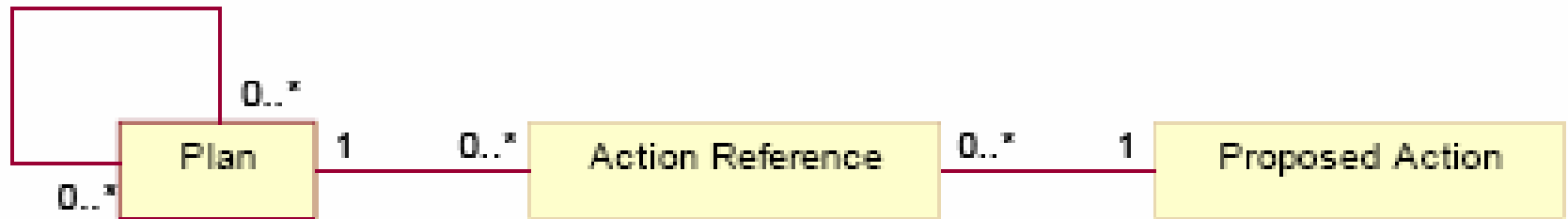
Plans as Actions and Compound Actions
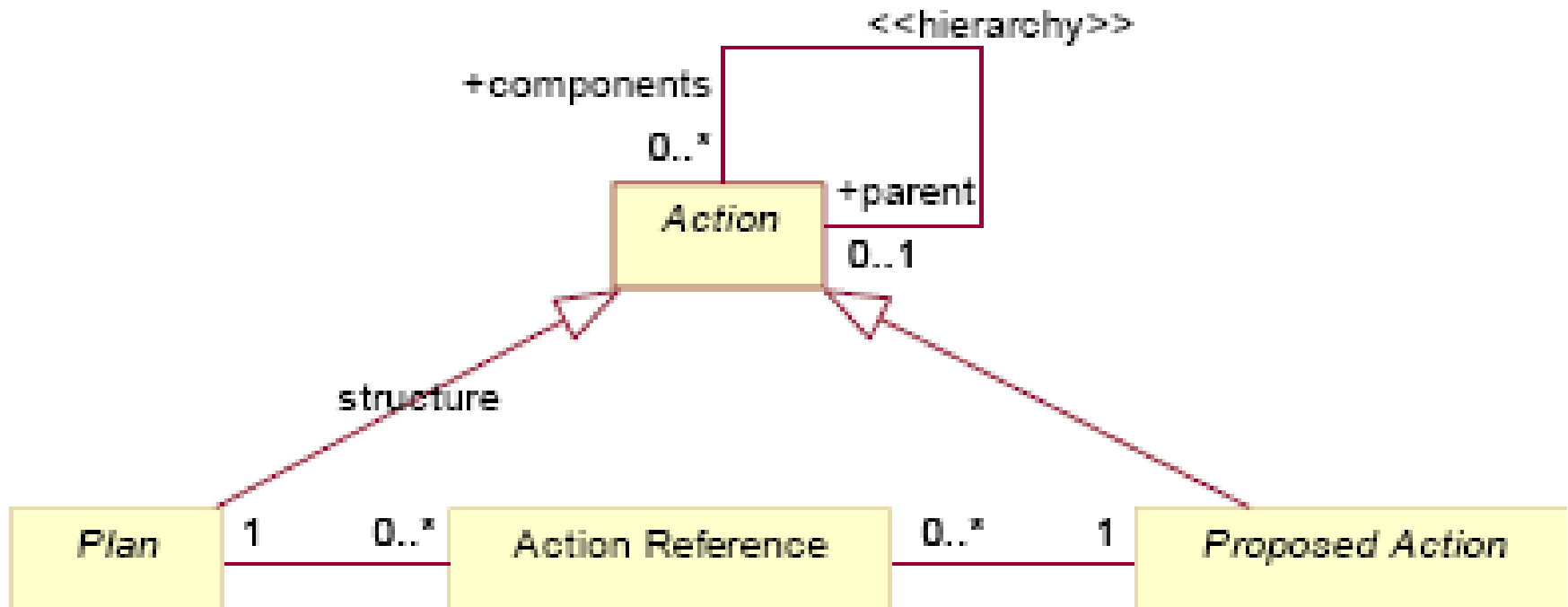
Structure for Protocols

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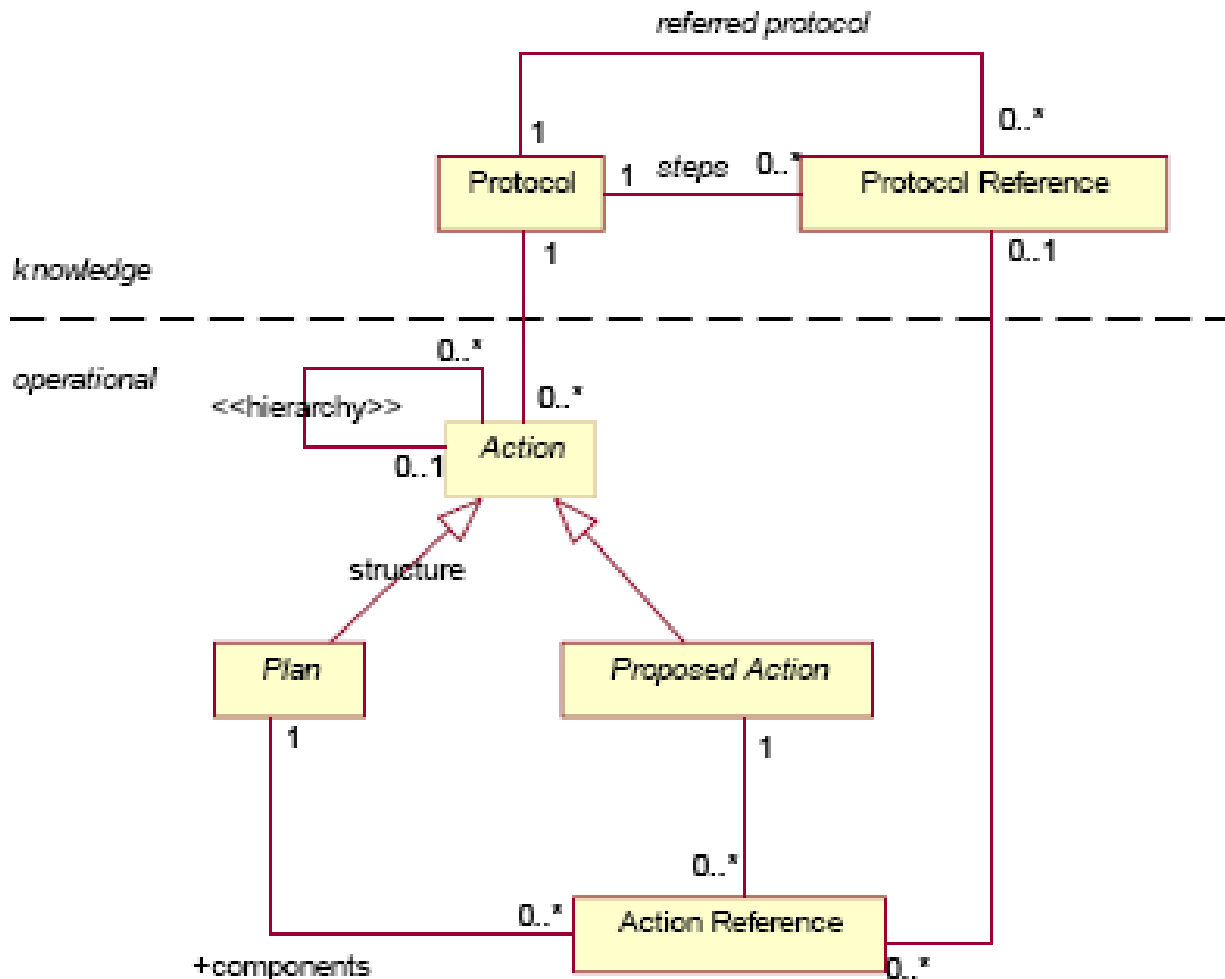
Plan as a **DAG** of Proposed Actions
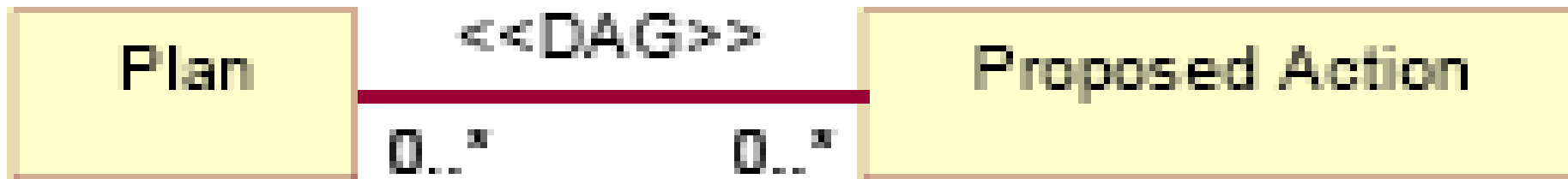
Protocol as DAG
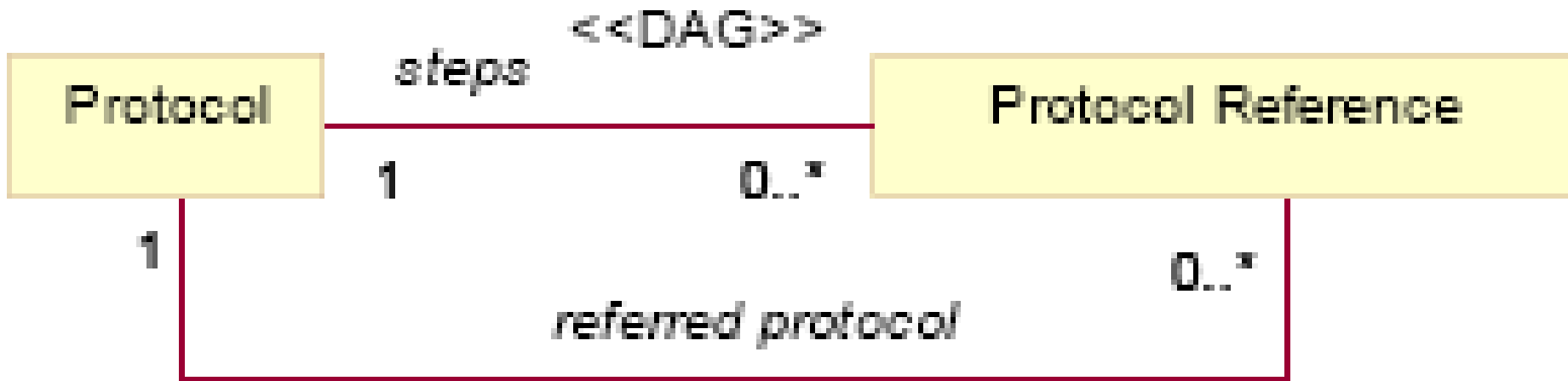
Action's Use of Resources
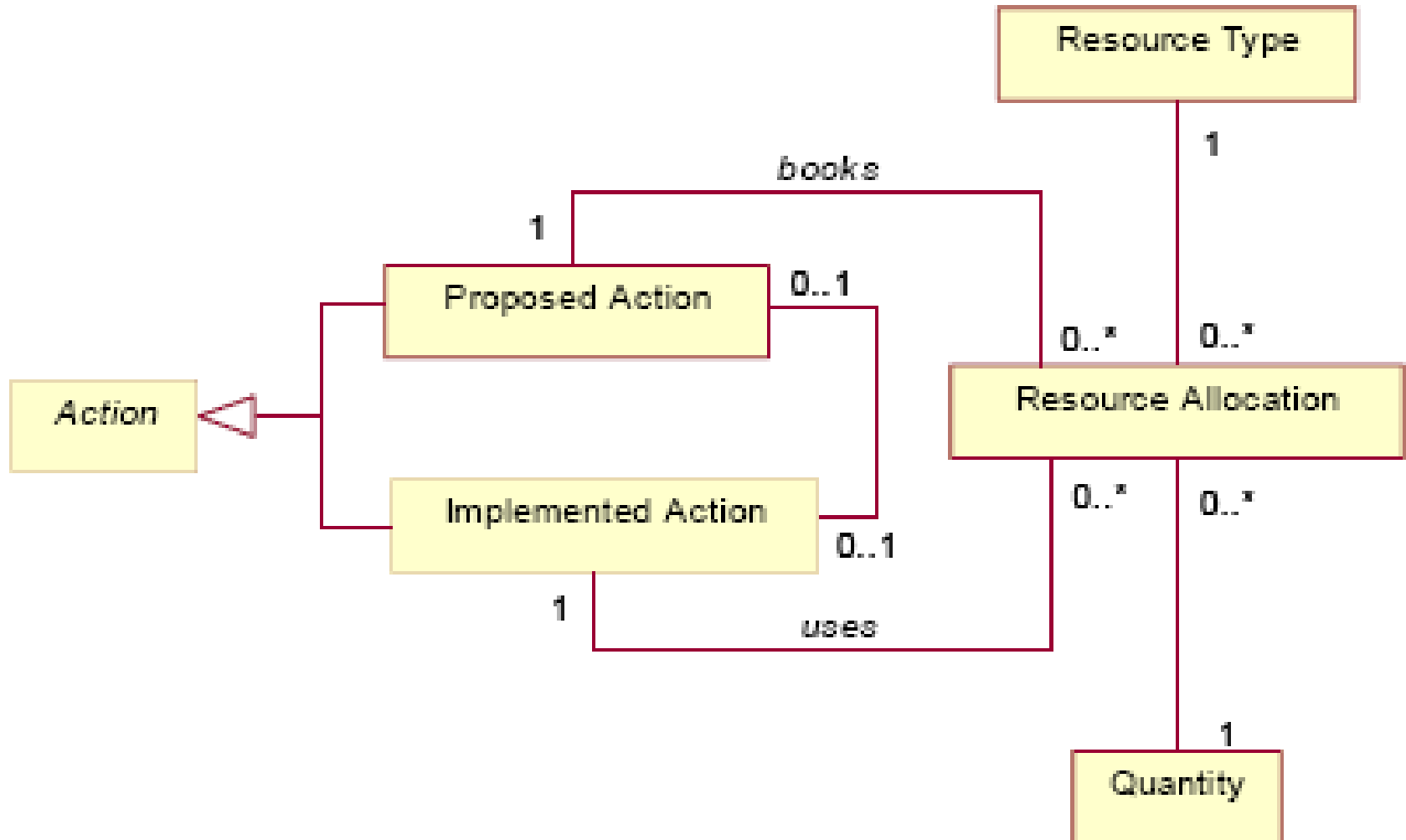
Ressource Allocation for Assets
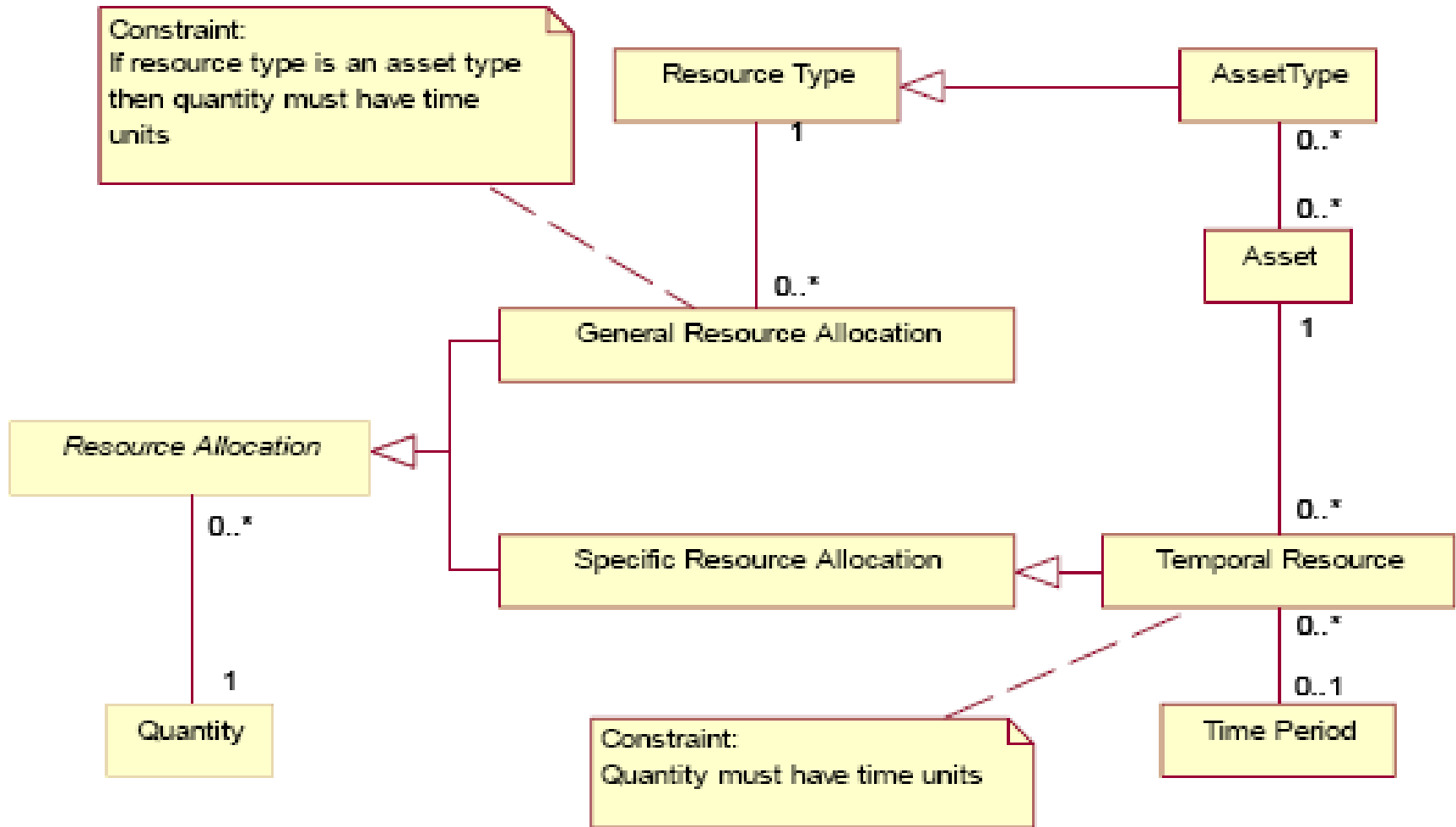
Allowing Specific Allocations of Consumables
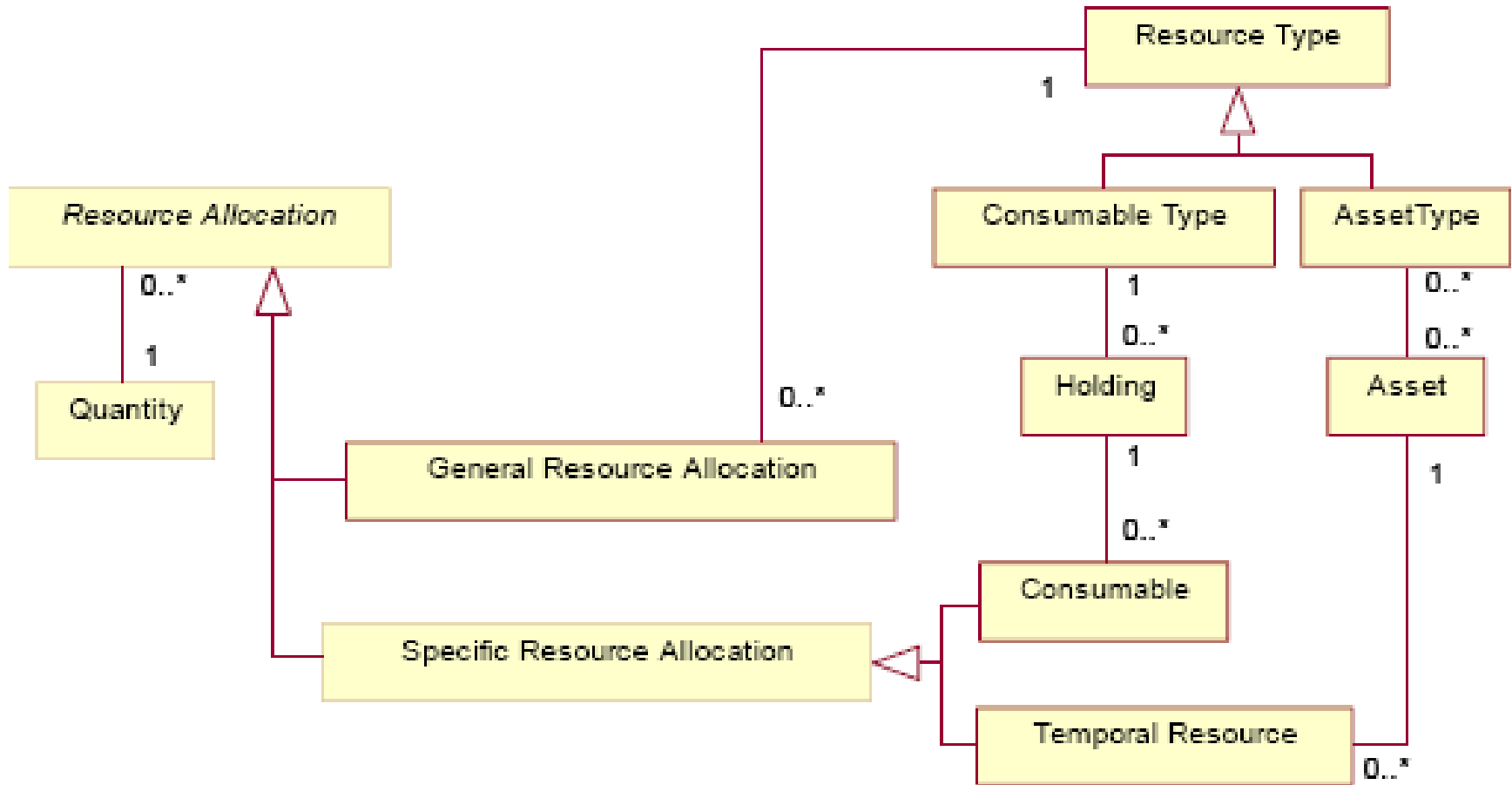
Links Between Observation, Plan, and Action
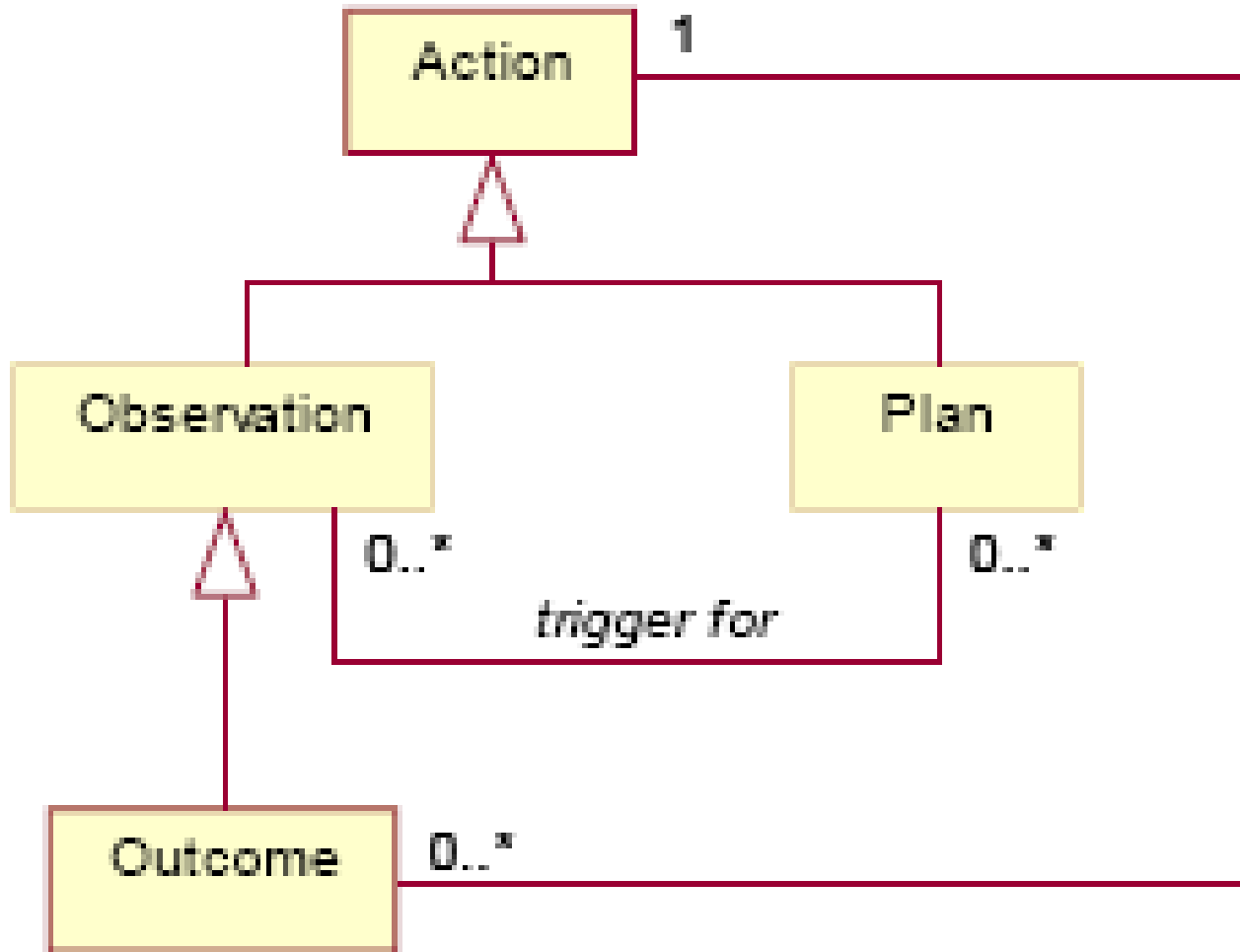
The Use of Start and Outcome Functions at the Knowledge Level
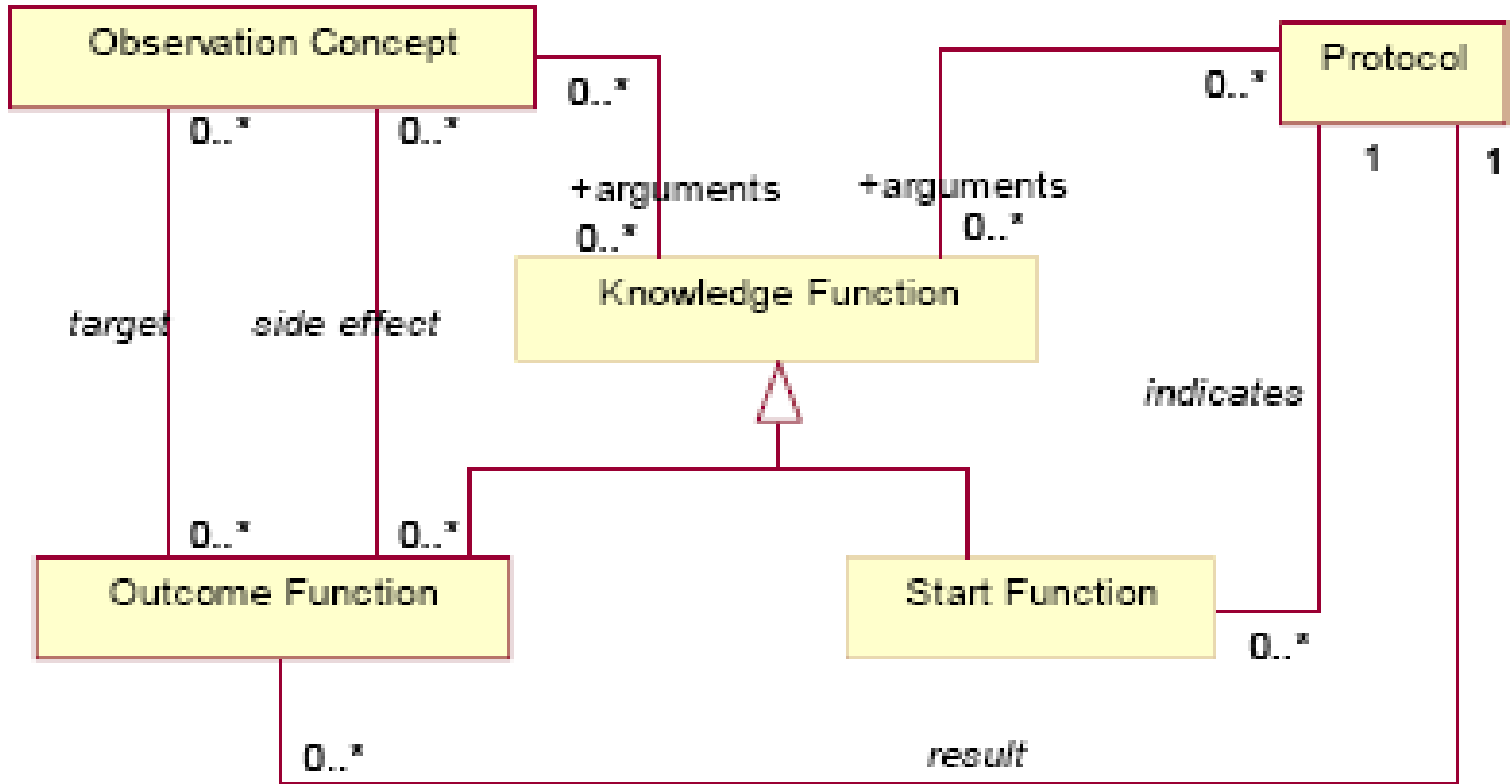
Trading

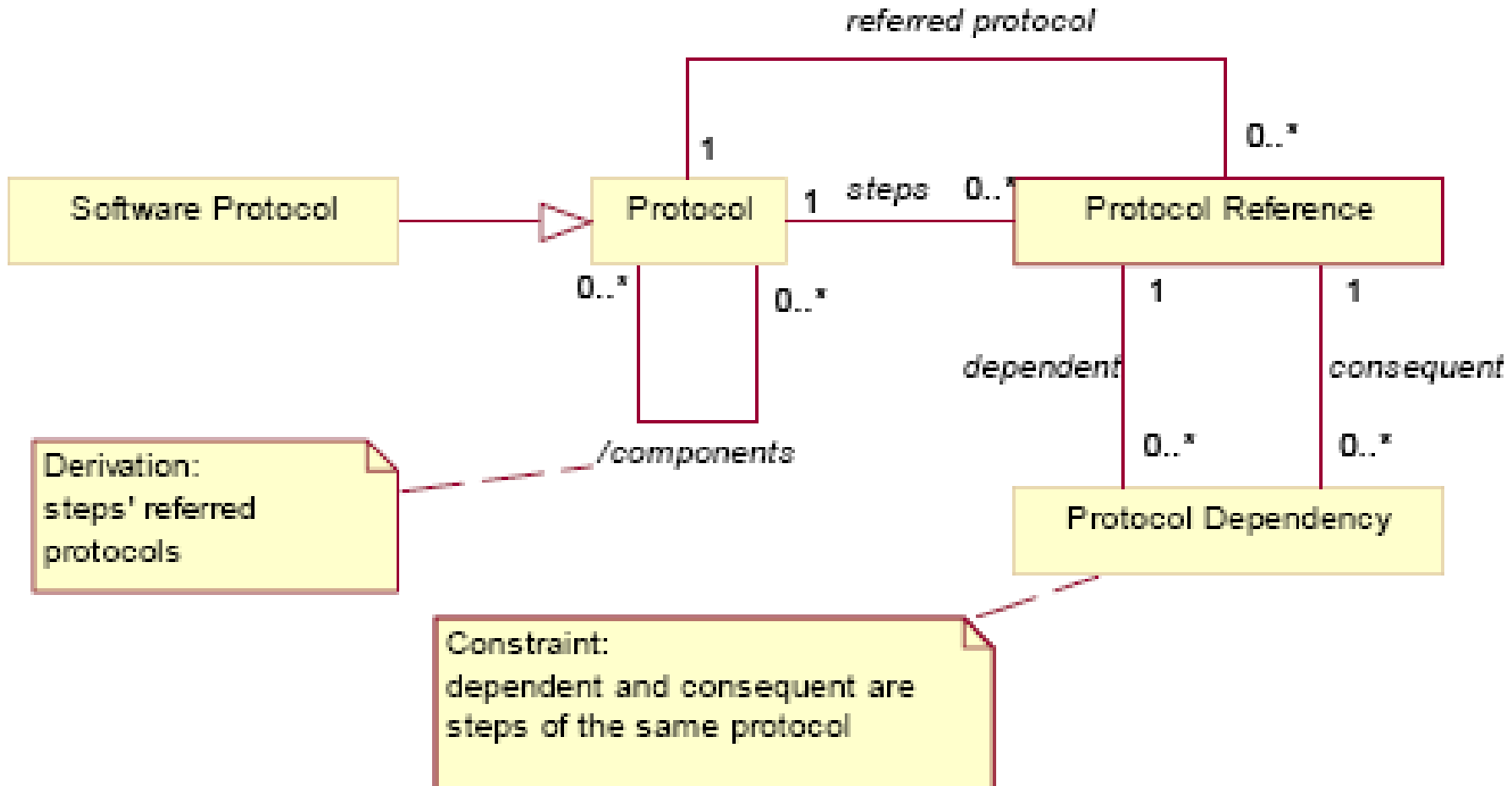Trade is about buying and selling
Trade is described by a contract
Contracts can be grouped into portfolios
Prices of goods change according to whether you look at them
from selling and buying point of view – quotes
Rapid changes can be captured by scenarios

Contract
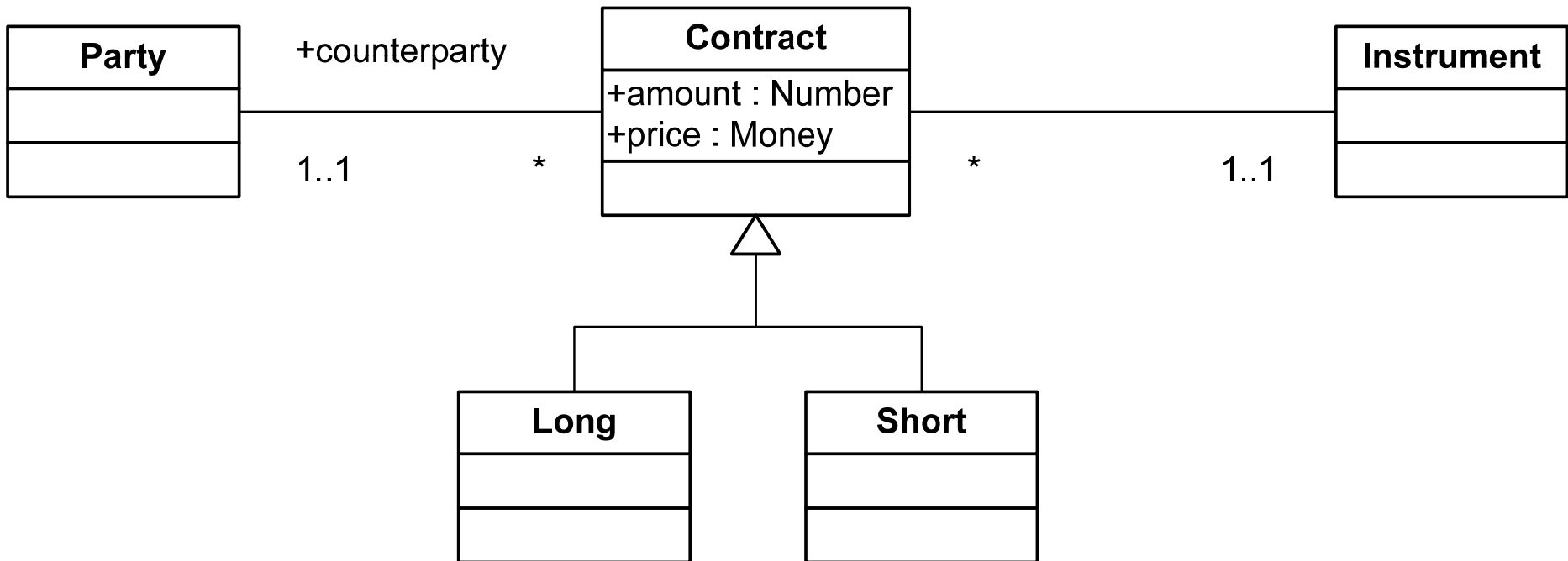
Buyers and Sellers and Relationship

```
                              +short

 ┌─────────────┐                      ┌─────────────────────────┐                    ┌──────────────────┐
 │    Party    │                      │        Contract         │                    │    Instrument    │
 ├─────────────┤          +Long       ├─────────────────────────┤                    ├──────────────────┤
 │             │    1..1              │ +amount : Number        │                    │                  │
 ├─────────────┤                      │ +price : Money          │        *      1..1 ├──────────────────┤
 │             │                      ├─────────────────────────┤                    │                  │
 └─────────────┘    1..1         *    └─────────────────────────┘                    └──────────────────┘
```
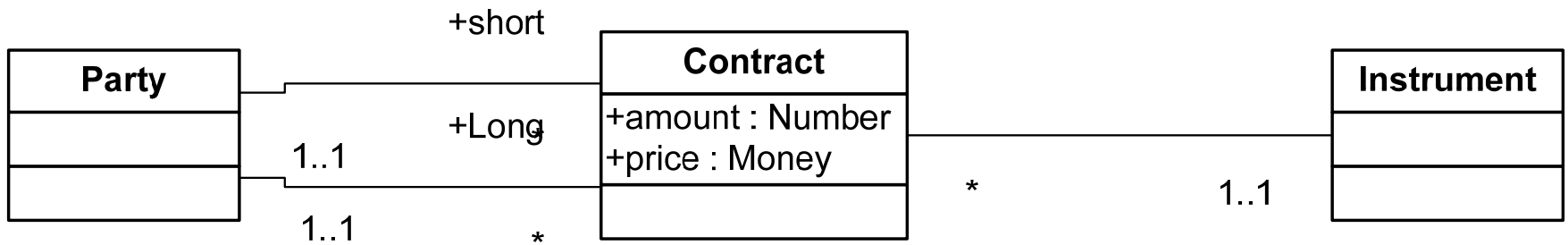
Counterparty and Primaryparty

Party mappings

«derived»

1..1   *   +short

+counterparty

Party   **Contract**   **Instrument**

1..1 +primaryparty

1..1   +amount : Number

+price : Money

1..1

«derived»   *   *

1..1   *   +long   «derived»

«derived»   **Short**

Long

Portfolio

Portfolio
-Value(scenario)

?
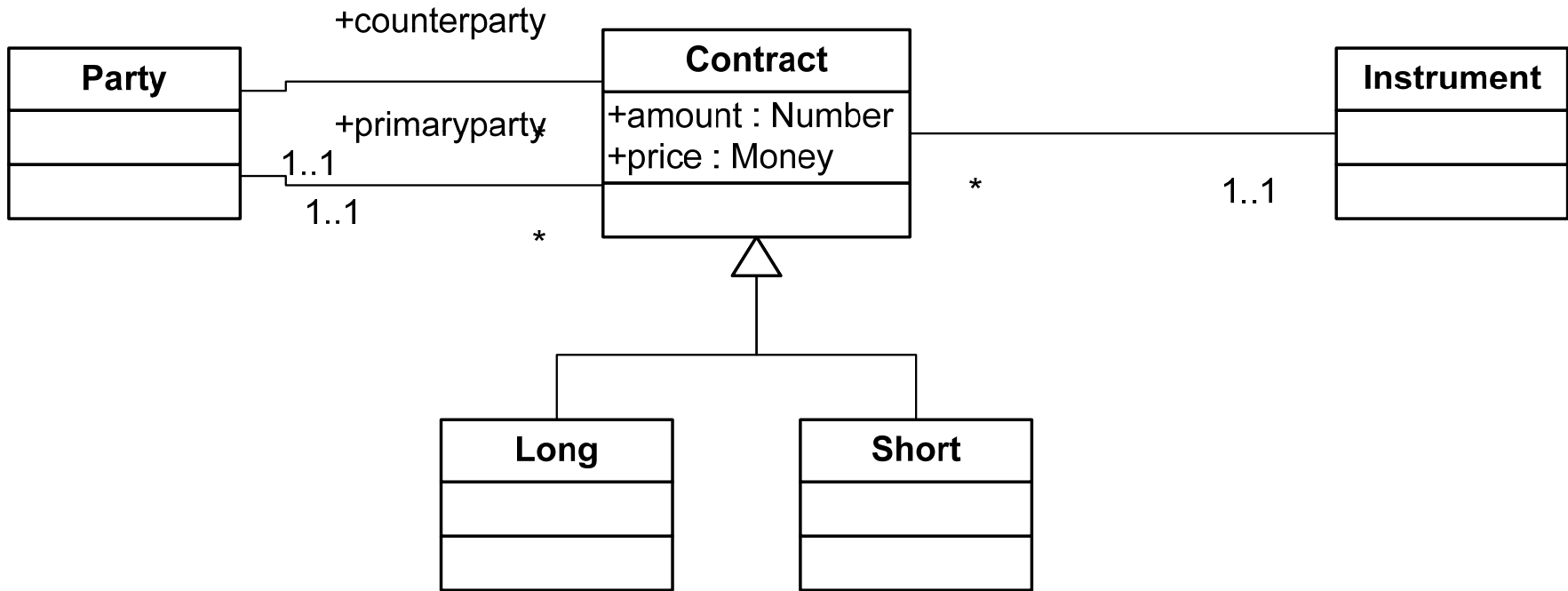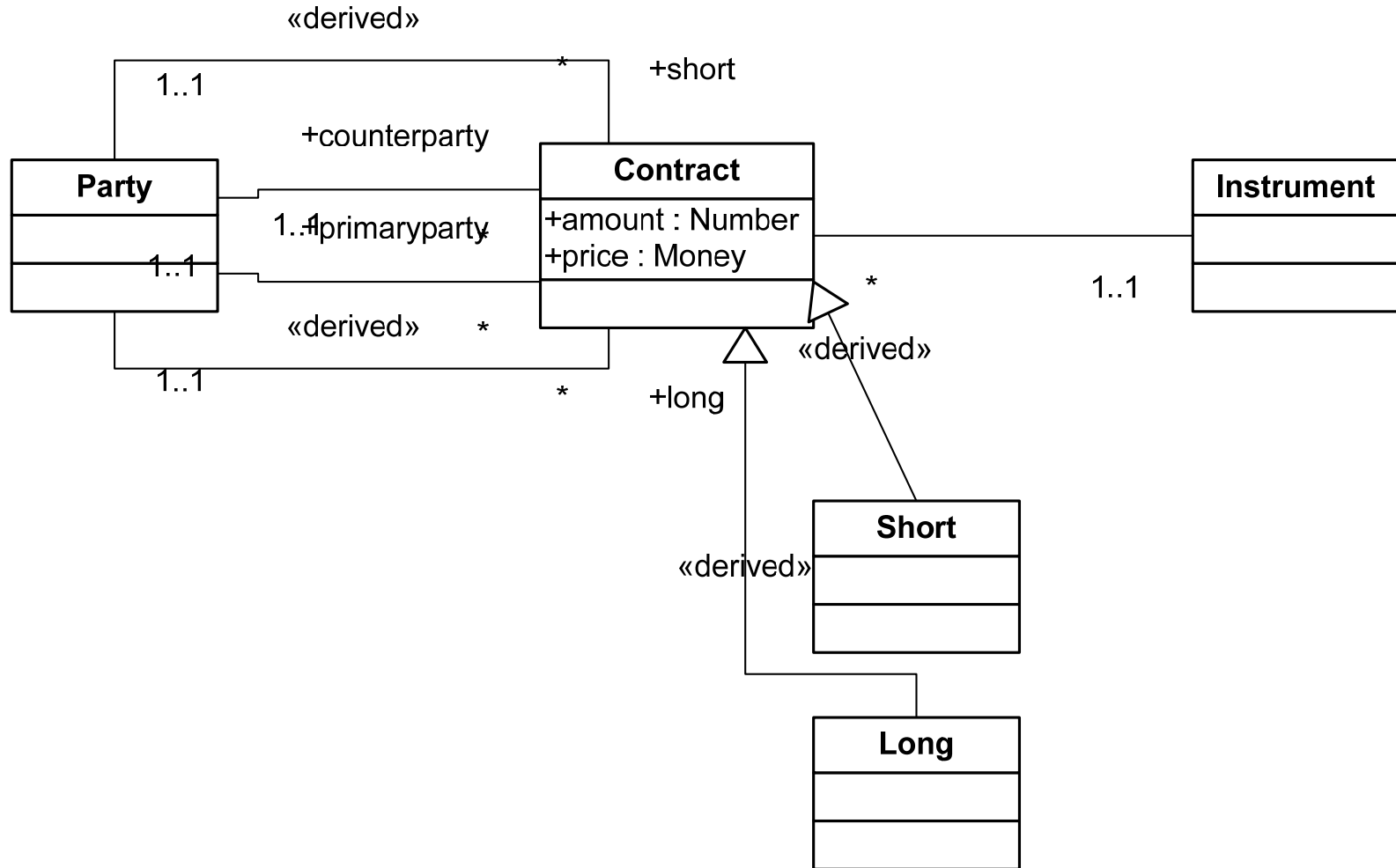
Contract
-Value(scenario)

*
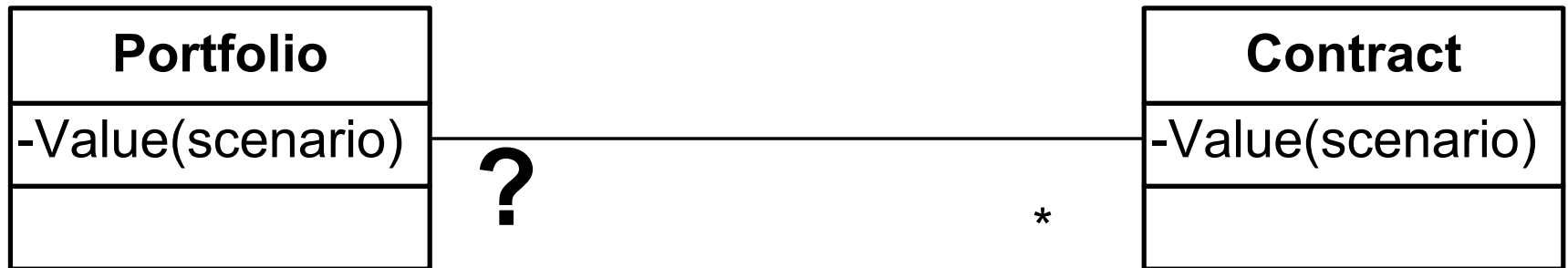
Dynamic portfolios with filters

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
-Value(scenario)

«derived»

Contract
-Value(scenario)

*   *

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

AALBORG UNIVERSITY

Several filter types

The diagram contains the following classes and relationships:

Date

ContractSelector with multiplicities 0..1, 0..1 +start +end to Date

Instrument — * — * — **ContractSelector**

ContractSelector — +primaryparties — **Party**
ContractSelector — * +counterparties * — **Party**

ContractFilter (abstract)
- -isIncluded
- +selectContracts(in Collection)

HardCodedFilter
- -Value(scenario)

BooleanMethod (1..1 +filter)

Portfolio
- -Value(scenario)

Contract
- -Value(scenario)

«derived»

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

SetFilter

SetOperation
{documentation = Instances: Union, Intersection, Negation}

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