Patterns in Software Engineering

Lecturer: Raman Ramsin

Lecture 18

Analysis Patterns

Part 1

Analysis Patterns

- First introduced by Fowler in 1997.
- “Analysis patterns are groups of concepts that represent a common construction in business modeling.“
  - An analysis pattern may be relevant to only one domain, or it may span many domains.

Analysis Patterns: Categories

- **Accountability:** Patterns for describing relationships that define responsibilities between parties.
- **Observations and Measurements:** Patterns for recording facts.
- **Referring to Objects:** Patterns of indexing to refer exactly to objects.
- **Inventory and Accounting:** Basic patterns for accounting, describing how a network of accounts can form an active accounting system.
- **Planning:** Patterns depicting the relationship between standard plans and one-off plans, and how to plan and record the use of resources.
- **Trading:** Patterns focusing on trading in situations where prices are fluid and we need to understand how these price changes affect the profits.

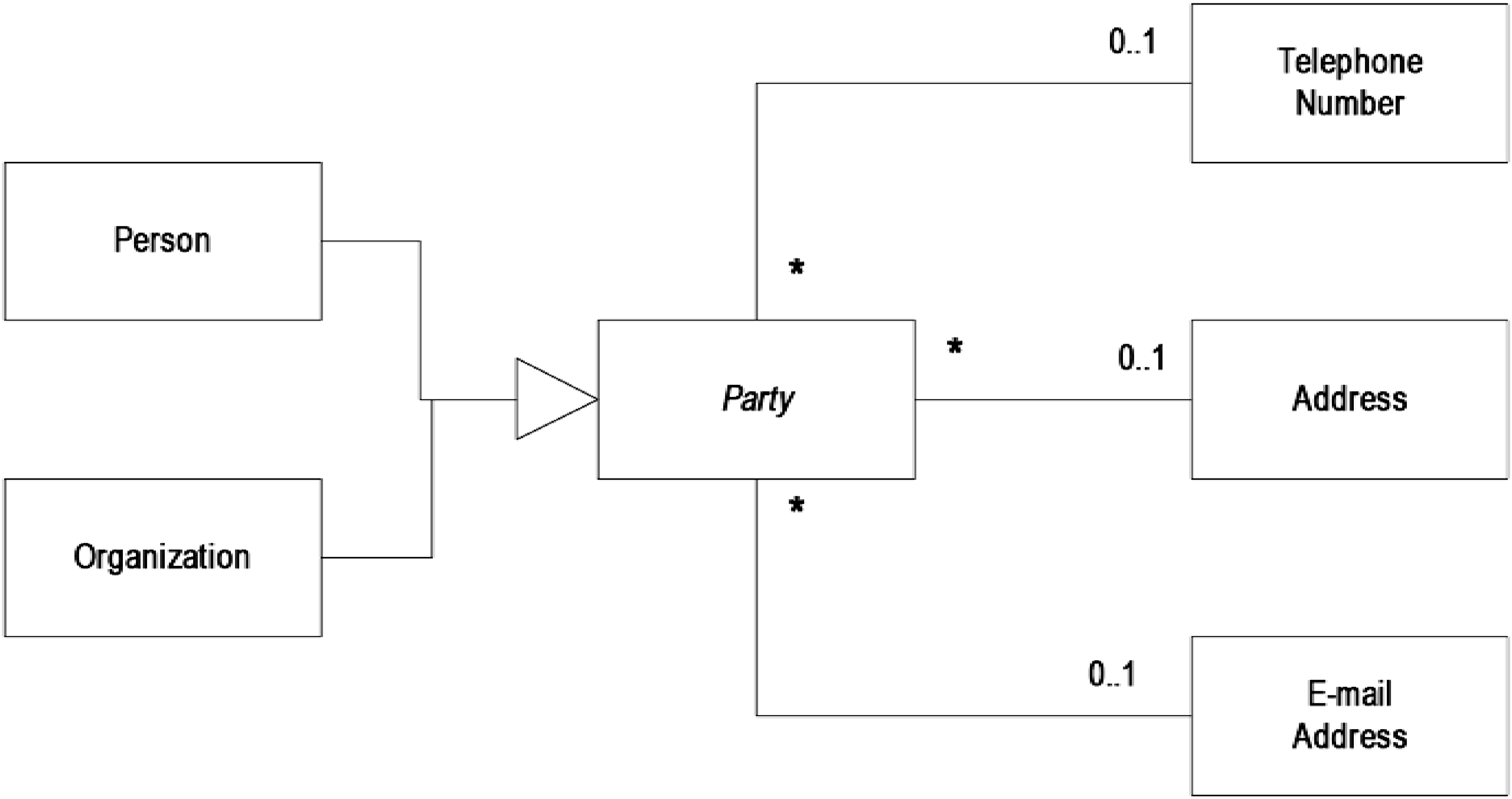
Supporting Patterns

- Fowler also provides several supporting patterns, which describe how to take analysis patterns and apply them:
  - **Layered Architecture for Information Systems**
  - **Patterns for Type Model Design Templates**
  - **Association Patterns**

Analysis Patterns: Accountability - *Party*

- **Problem:** People and organizational units have similar responsibilities.
- **Solution:** Create a type party as a supertype of person and organization.

Analysis Patterns: Accountability - *Party*

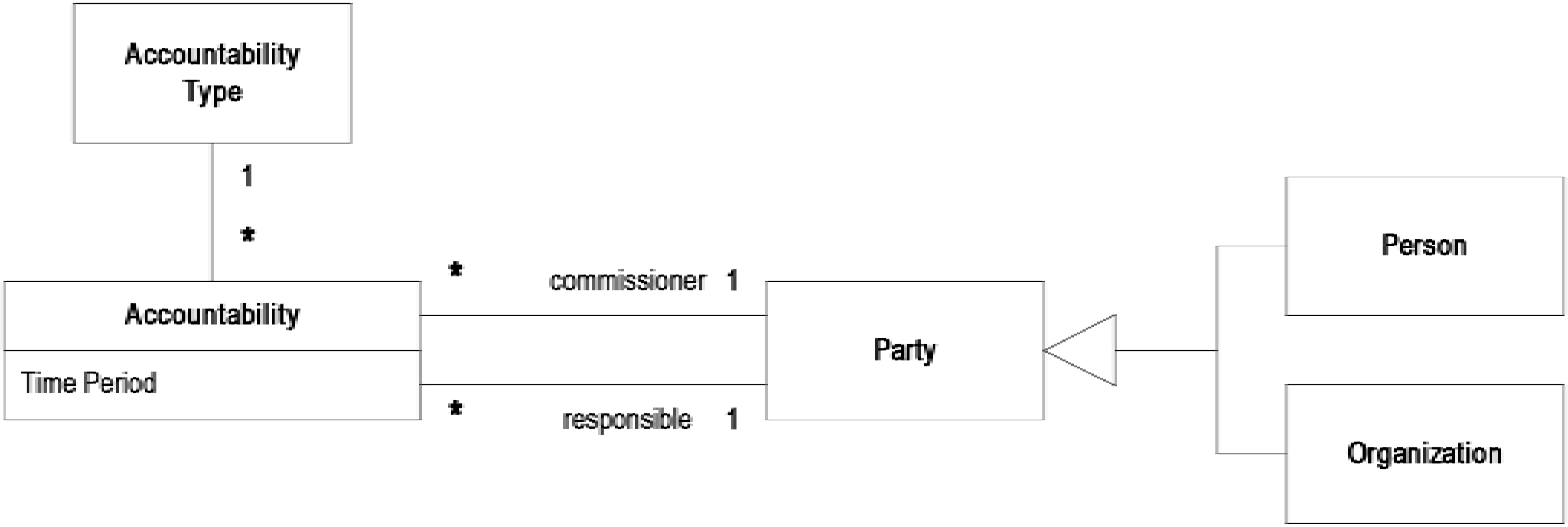


Analysis Patterns: Accountability - *Accountability*

- **Problem:** Representing organization structures, employment, management, professional registration, and contracts with a similar structure.

- **Solution:** Create accountability as a directed relationship between two parties.
  - Give it an accountability type to represent the kind of relationship.

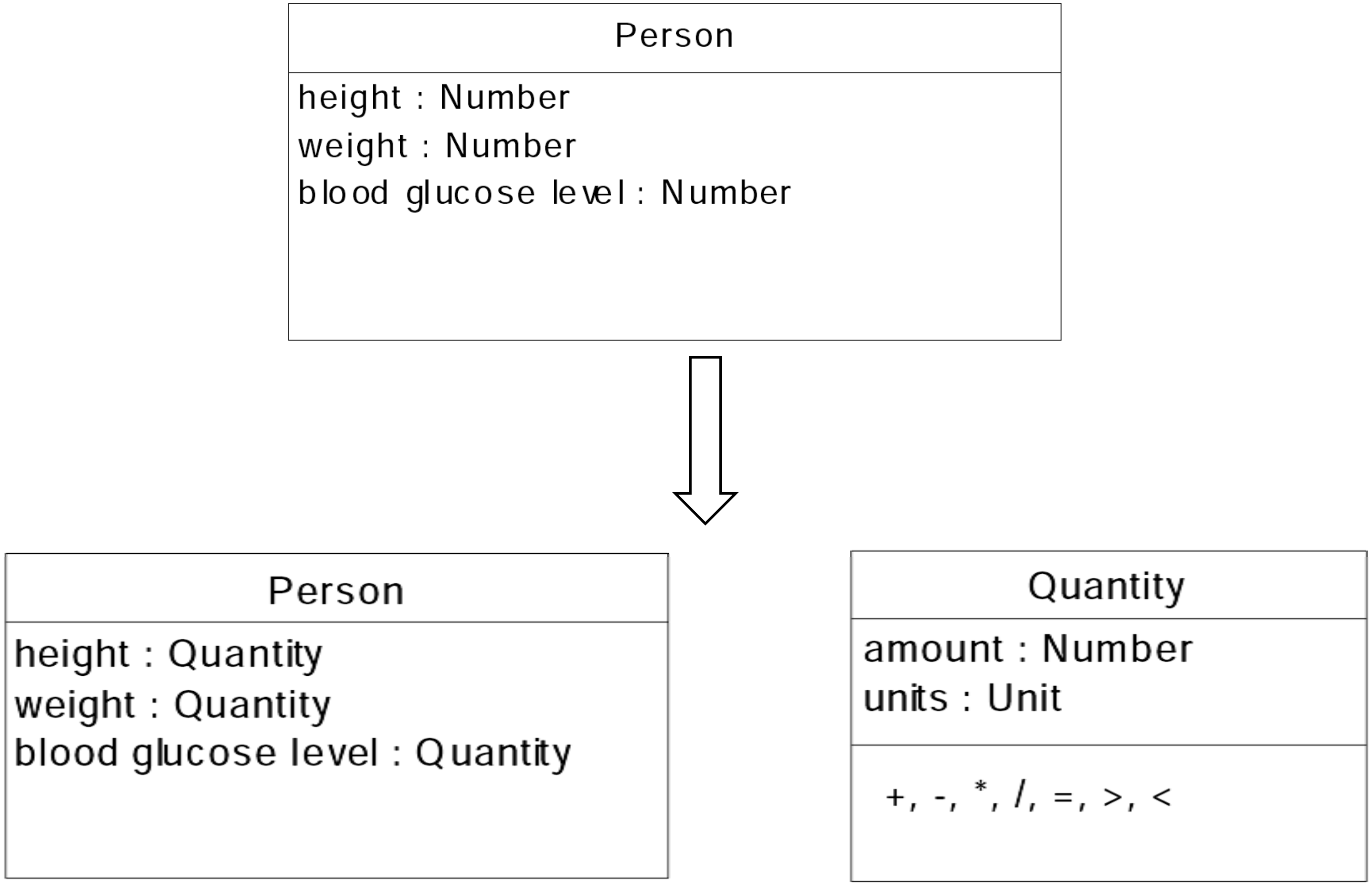
Analysis Patterns: Accountability - *Accountability*



Analysis Patterns: Observations and Measurements - *Quantity*

- **Problem:** Representing a value such as 6 feet or $5.
- **Solution:** Use a quantity type that includes both the amount and the unit.
  - Currencies are a kind of unit.

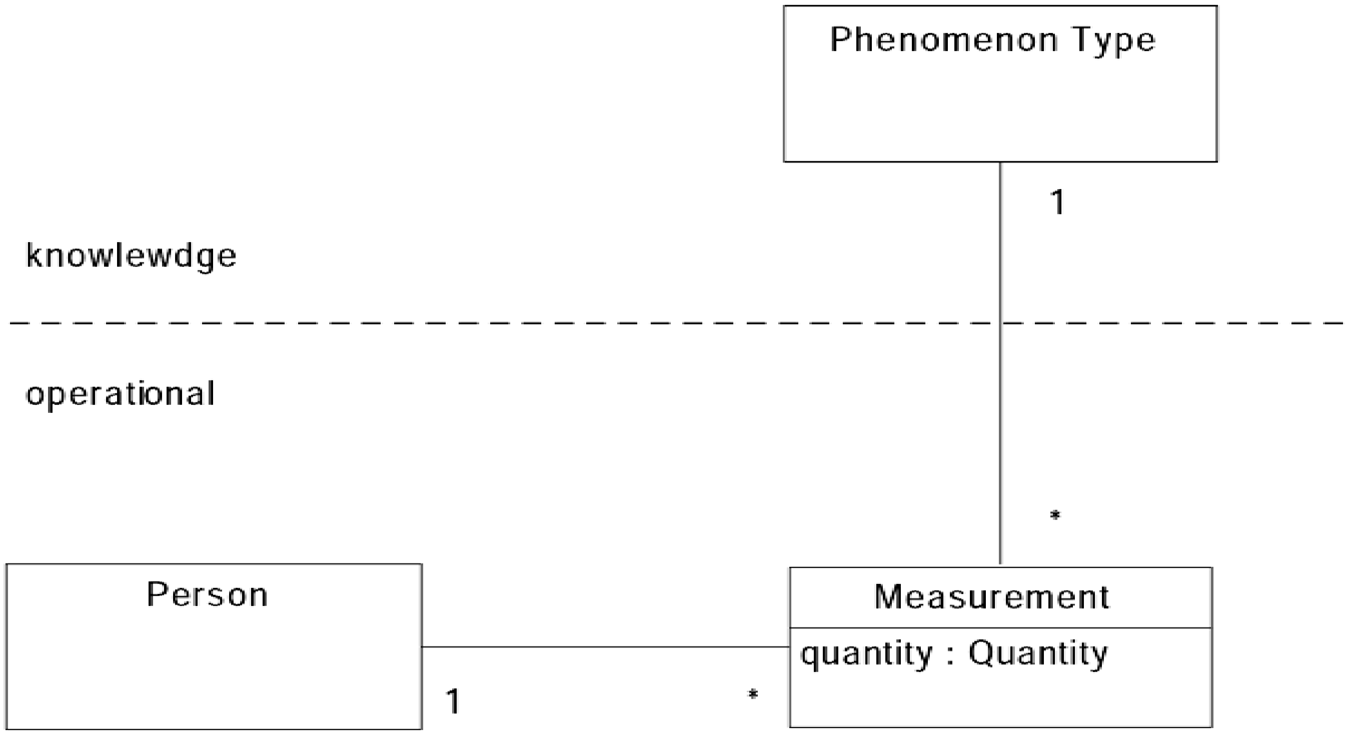
Analysis Patterns: Observations and Measurements - *Quantity*



Analysis Patterns: Observations and Measurements - *Measurement*

- **Problem:**
  - An object has a large number of quantity attributes.
  - Recording information about an individual measurement of an attribute.
  - Tracking changes in a value to an attribute over time.
- **Solution:** Create an object to represent the individual measurement.
  - This is linked to the object being measured and to a phenomenon type that describes the kind of measurement being made.

Analysis Patterns: Observations and Measurements - *Measurement*

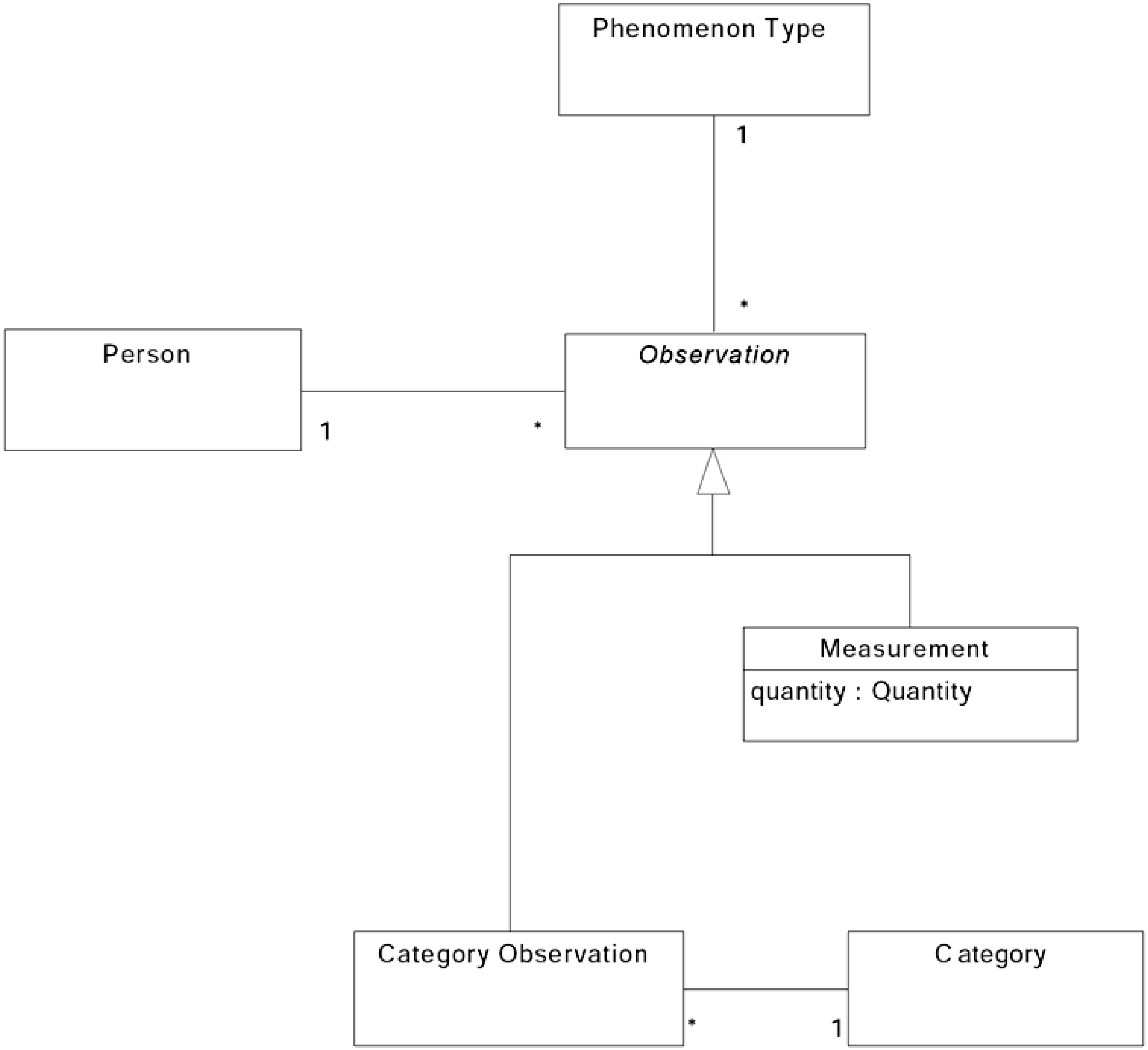


Analysis Patterns: Observations and Measurements - *Observation*

- **Problem:** Attributes are qualitative and thus cannot be measured with numbers.

- **Solution:** Create an observation type that links the object to a phenomenon.
  - Each phenomenon is a value for some phenomenon type.

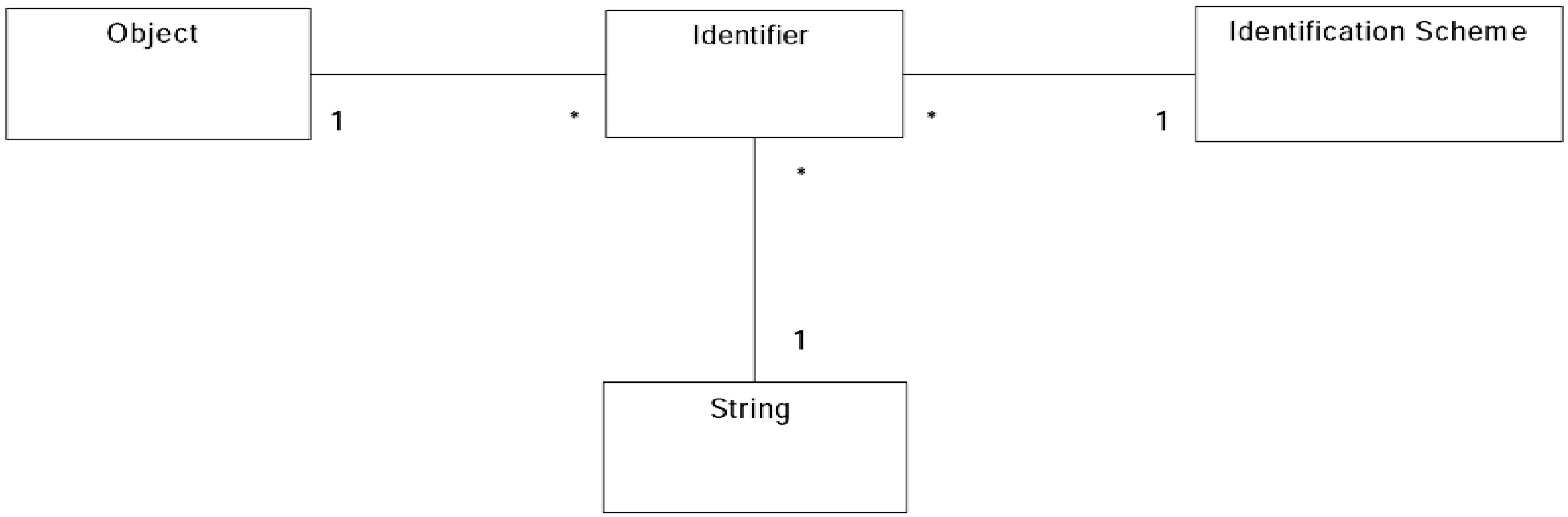
Analysis Patterns: Observations and Measurements - *Observation*



Analysis Patterns: Referring to Objects - *Identification Scheme*

- **Problem:** Ensuring an identification refers to only one object but different parties can refer to the object differently.
- **Solution:** Create identification schemes that contain identifiers, where each identifier refers to only one unit.
  - A party can use any identification scheme.

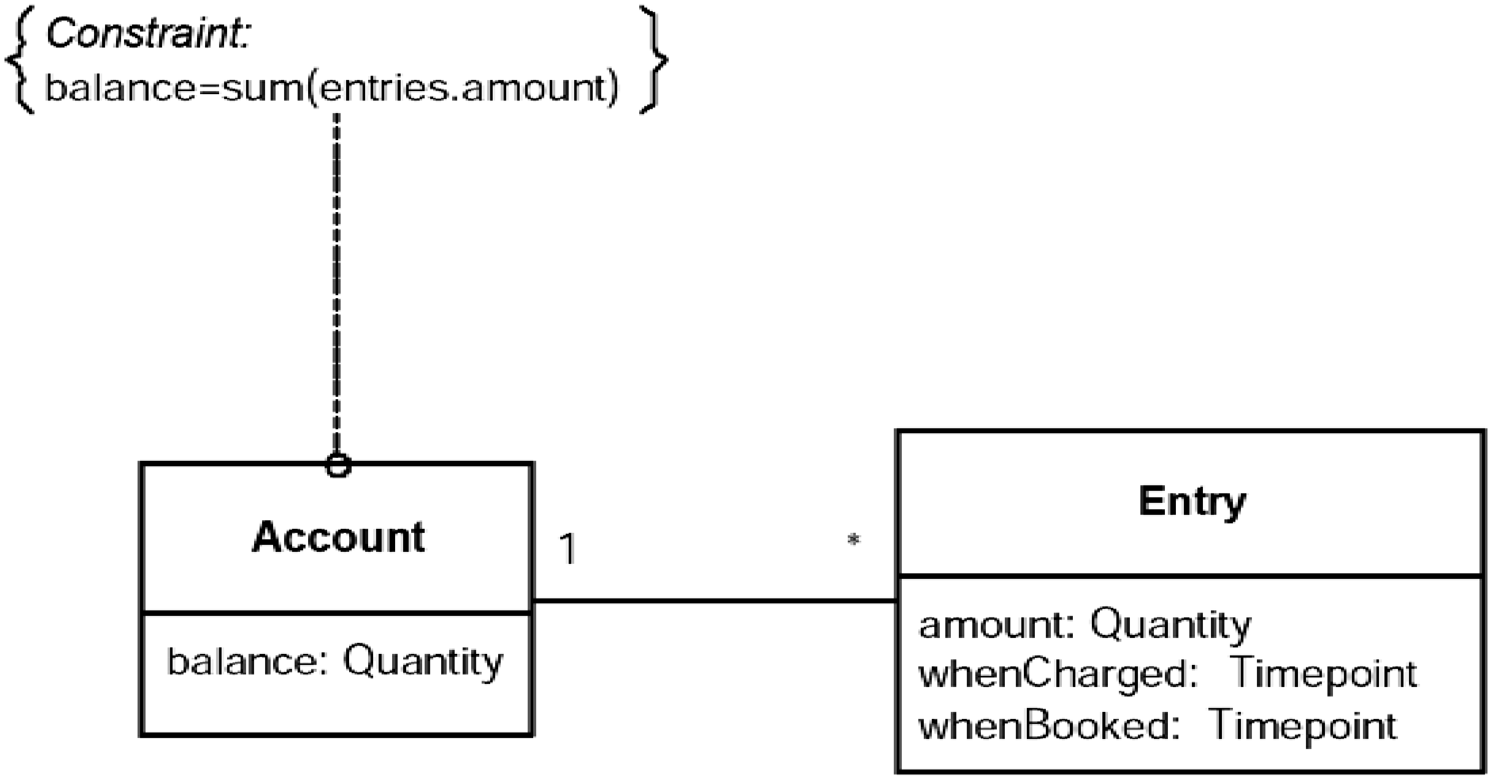
Analysis Patterns: Referring to Objects - *Identification Scheme*



Analysis Patterns: Inventory and Accounting - *Account*

- **Problem:** Recording a history of changes to some quantity.
- **Solution:** Create an account. Each change is recorded as an entry against the account.
  - The balance of the account gives its current value.

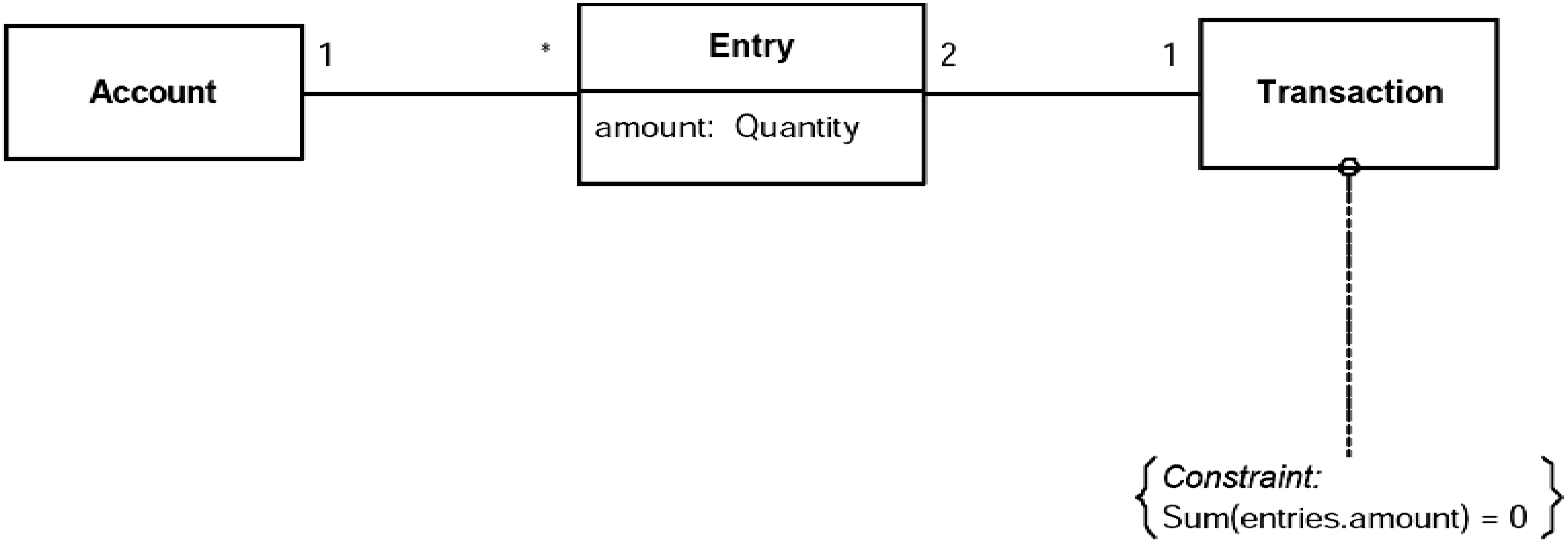
Analysis Patterns: Inventory and Accounting - *Account*



Analysis Patterns: Inventory and Accounting - *Transaction*

- **Problem:** Ensuring that nothing gets lost from an account.
- **Solution:** Use transactions to transfer items between accounts.

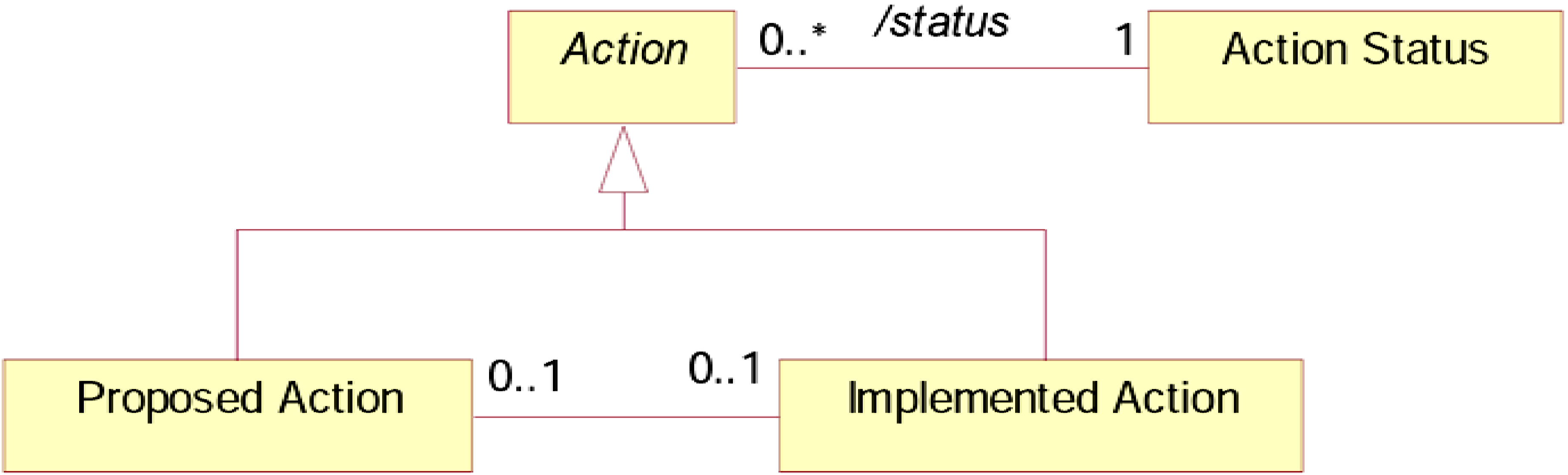
Analysis Patterns: Inventory and Accounting - *Transaction*



Analysis Patterns: Planning - *Proposed and Implemented Action*

- **Problem:** Representing both what you intended to do and what you did.
- **Solution:** Use separate objects for the proposed and implemented actions.

Analysis Patterns: Planning - *Proposed and Implemented Action*

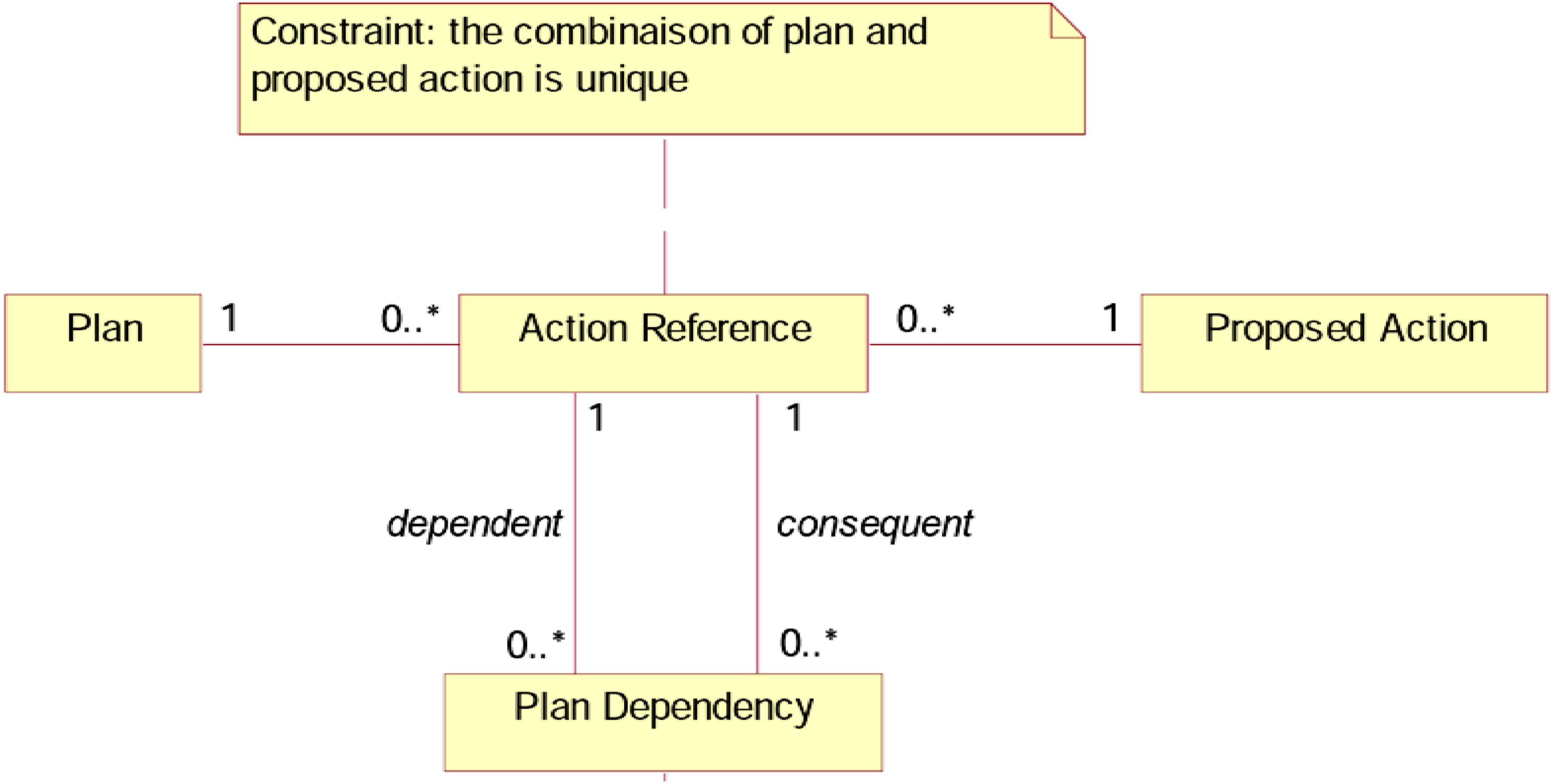


Analysis Patterns: Planning - *Plan*

- **Problem:**
  - Recording a group of proposed actions that you intend to perform together.
  - Representing the dependencies among actions.
  - Allowing different people to coordinate each other's plans.

- **Solution:** A plan is a collection of proposed actions linked by dependencies.
  - Several parties can have different plans that refer to the same proposed action.
  - Use separate objects for the proposed and implemented actions.

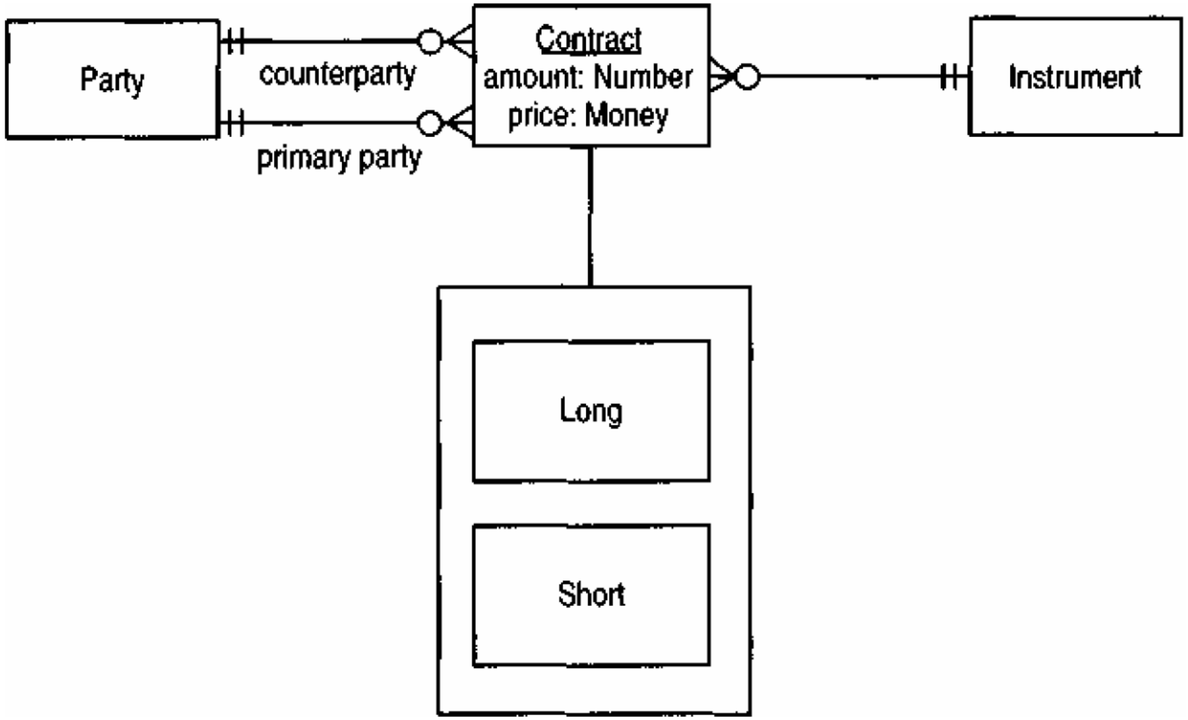
Analysis Patterns: Planning - *Plan*



Analysis Patterns: Trading - *Contract*

- **Problem:** Recording deals from the perspective of both the buyer and the seller.
- **Solution:** Use a contractor with both buying and selling parties.

Analysis Patterns: Trading - *Contract*



Reference

- Fowler, M., *Analysis Patterns: Reusable Object Models,* Addison-Wesley, 1997.