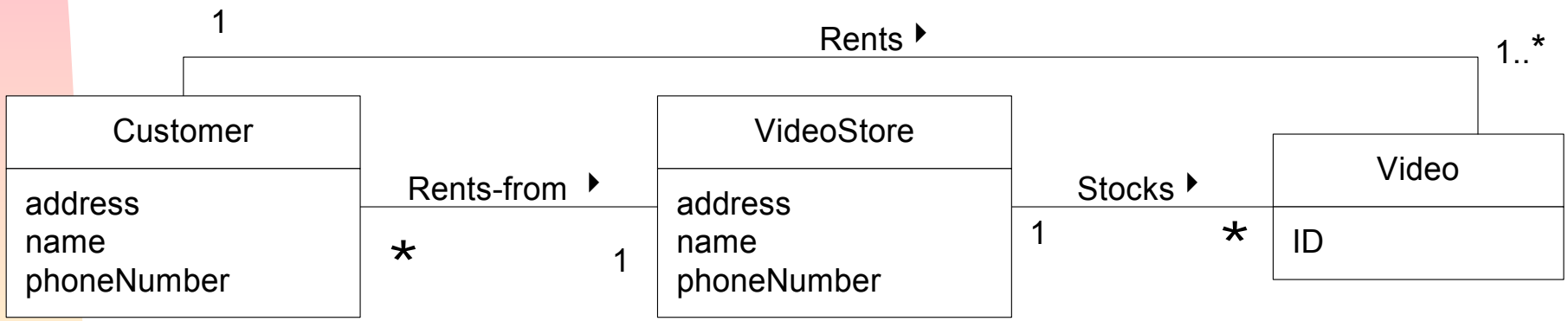
The Domain Model

Introduction

- Partitions and illustrates the important domain concepts.
- A classic object-oriented analysis activity.
- What are the objects of interest in the this domain?
  - their attributes?
  - their relationships?
- IMPORTANT: Not software objects, but a “visual dictionary” of domain concepts.

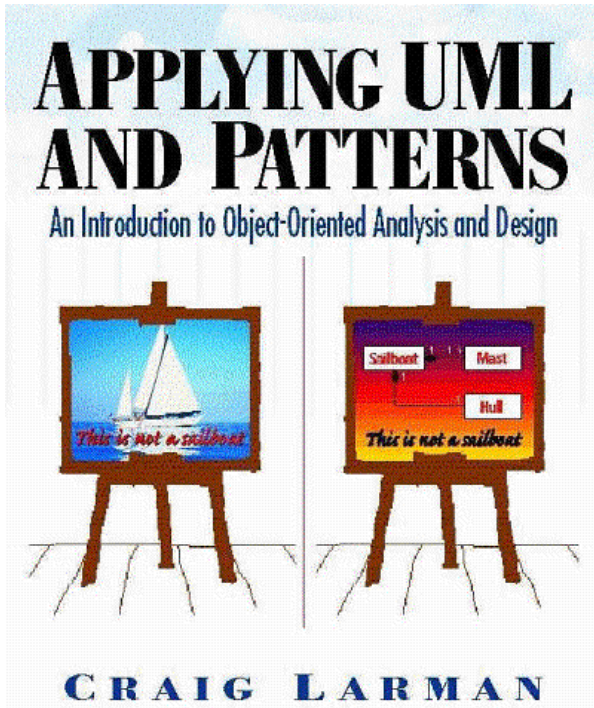
A Domain Model Does Not Represent Software Objects

- **A model of domain concepts, not of software objects.**
  - **A "visual dictionary" of important words in the domain.**
- **Uses UML *static structure diagram* notation.**



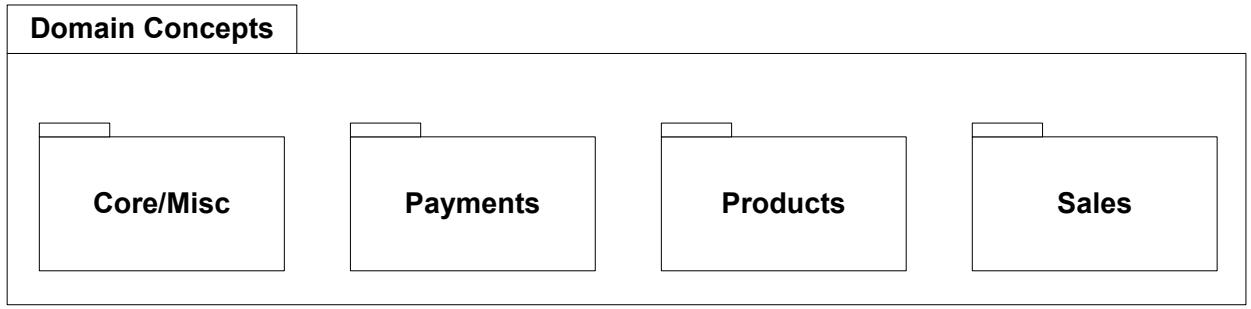
Terminology

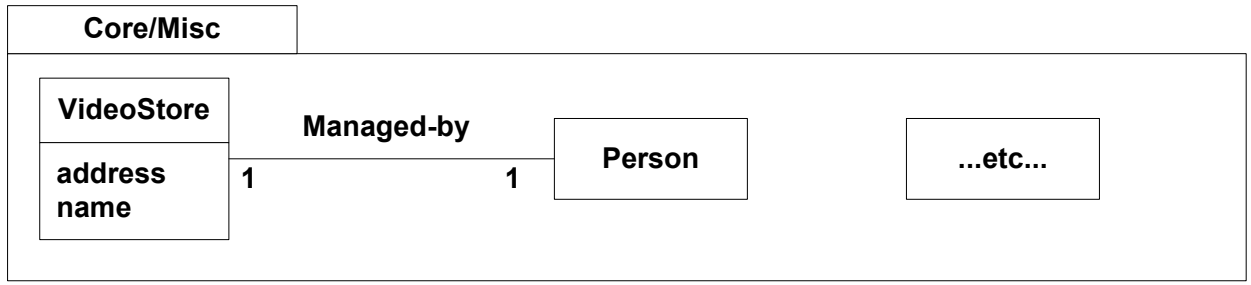
- UP *domain model* = book *conceptual model*

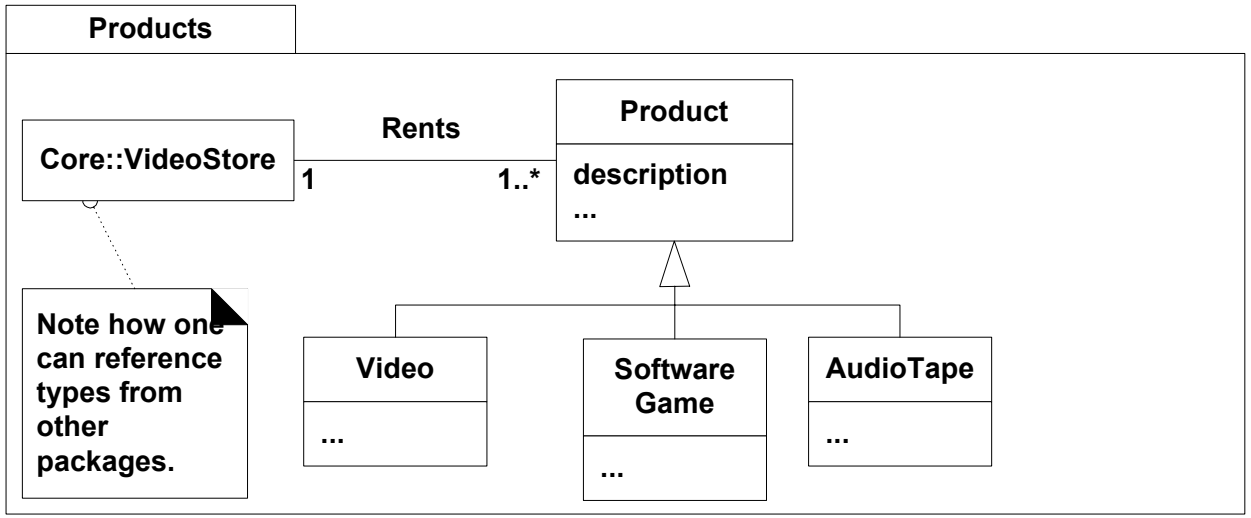


Partitioning the Domain Model

- Although this is conceptually accurate, a person will not actually draw package boxes around groups as in this example. Rather, a CASE tool will allow "drill down".



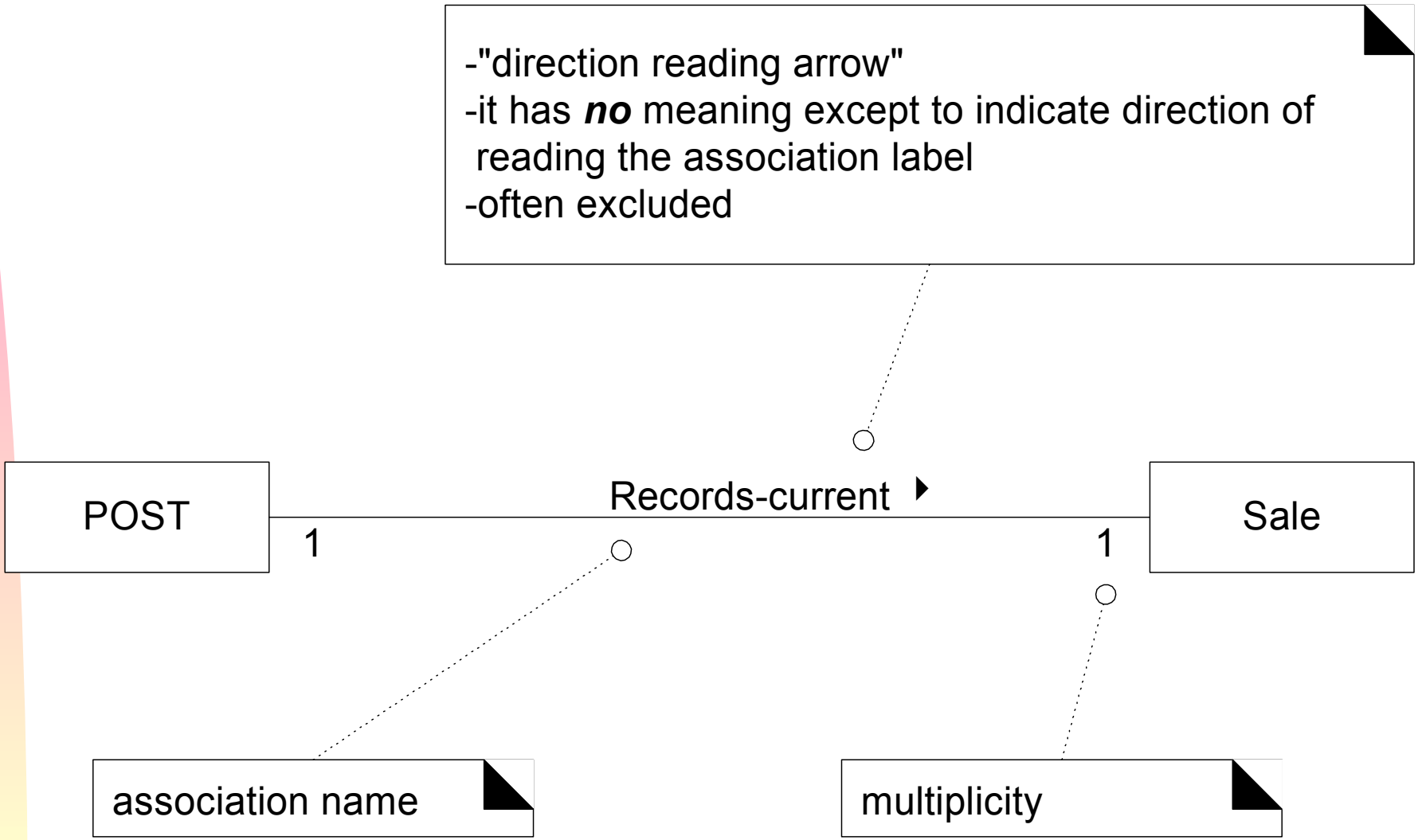




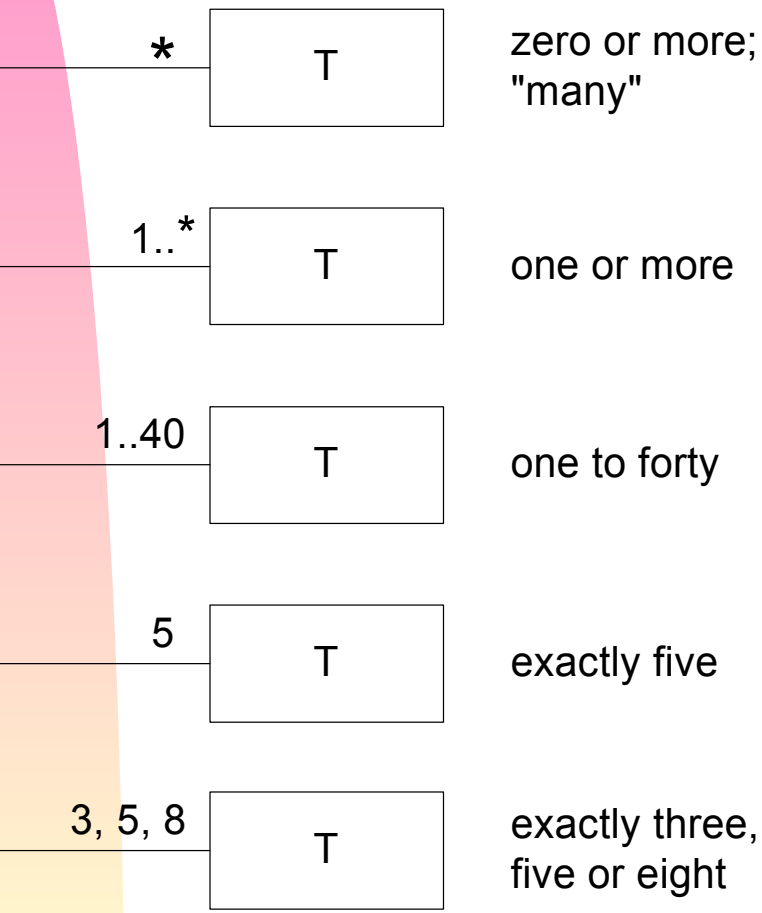
Finding Domain Concepts

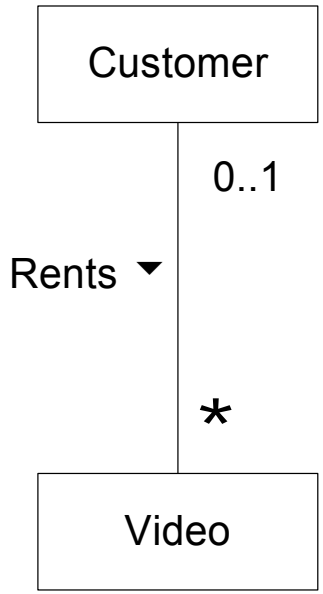
- Candidate lists
- "Abbott" Analysis
- Existing analysis patterns
  - Analysis Patterns, Fowler
  - Data Model Patterns, Hay
  - The Data Model Resource Book, Silverston

Associations



Multiplicity

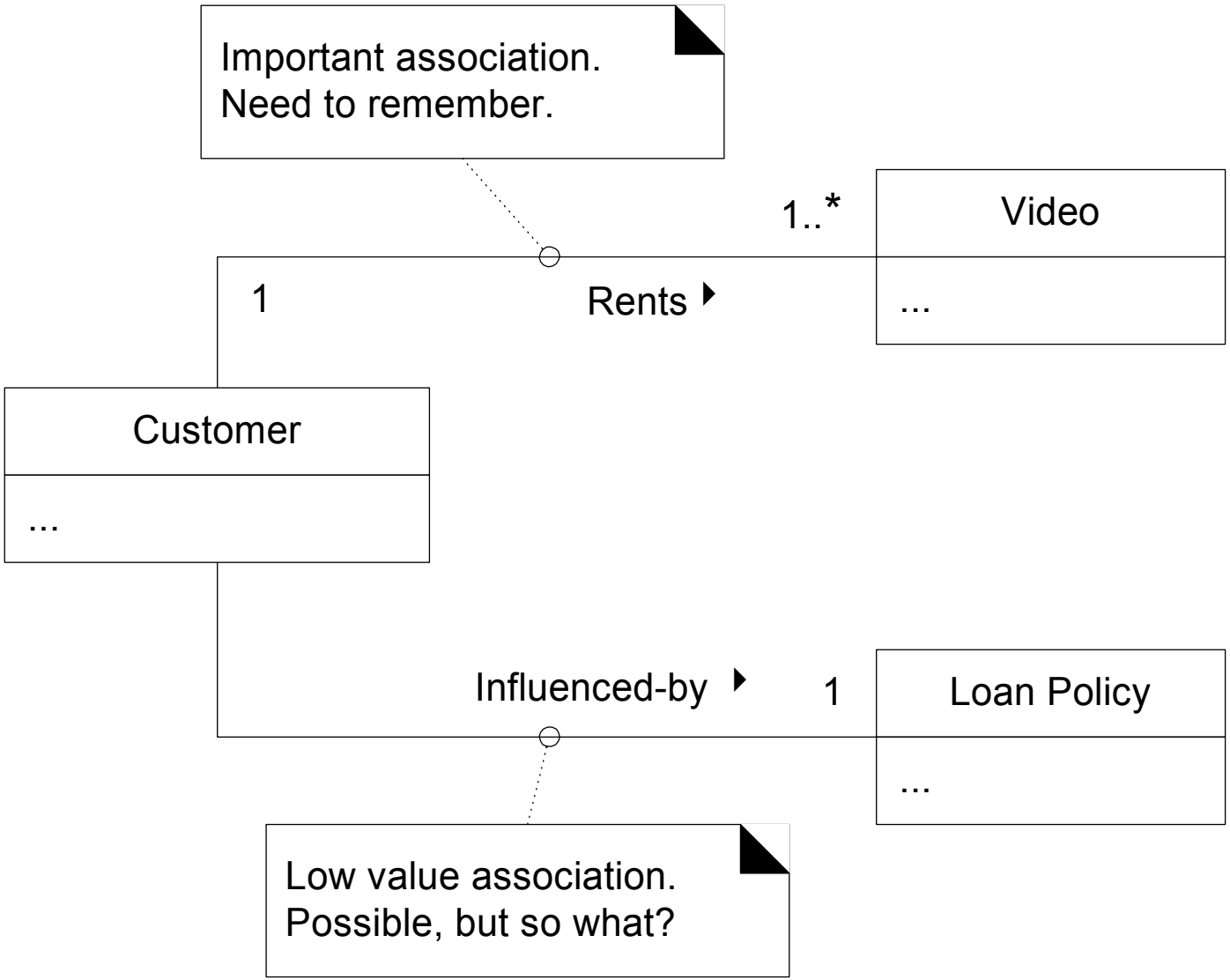




One instance of a Customer may be renting zero or more Videos.

One instance of a Video may be being rented by zero or one Customers.

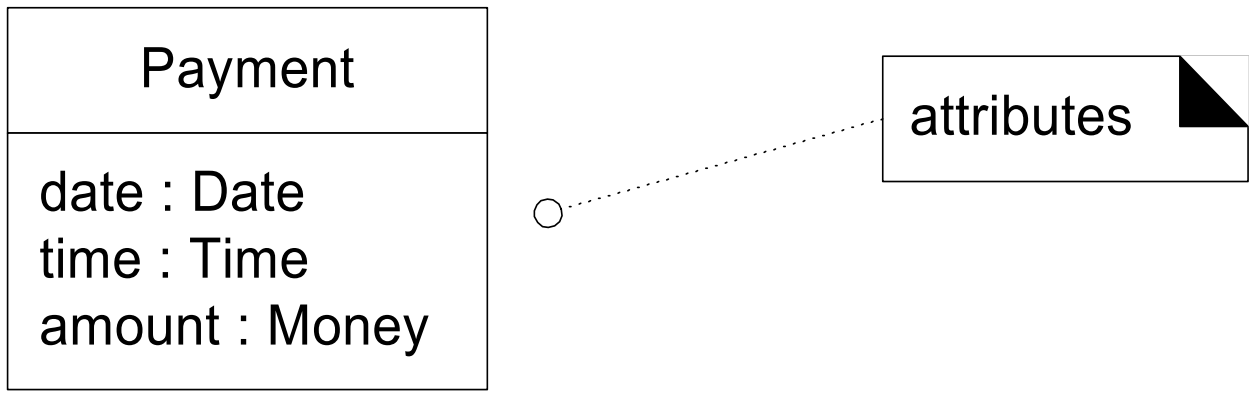
Focus on Important Associations



Attributes

- **Show only “simple” relatively primitive types as attributes.**

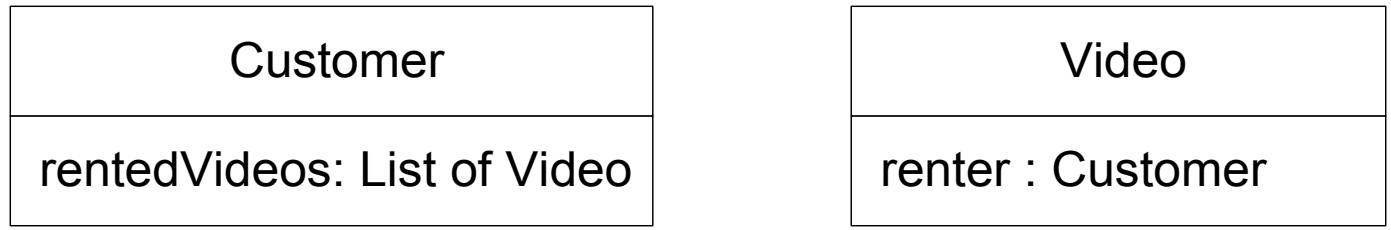
- **Connections to other concepts are to be represented as associations, not attributes.**



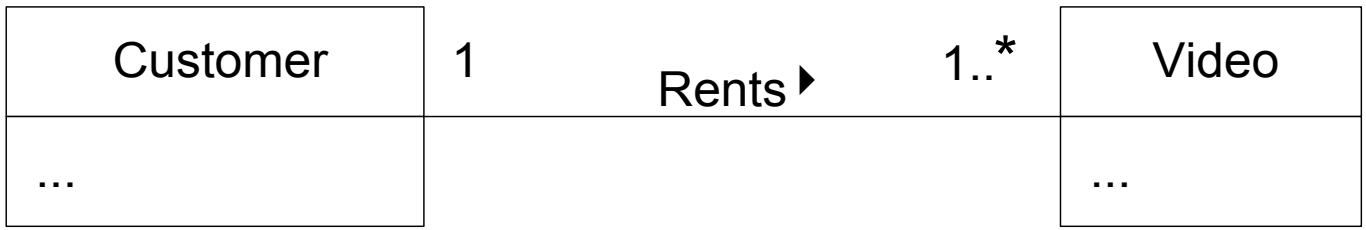
Do Not Use Attributes To Relate Concepts

- **Why not?**

Worse



Better



An Example