Classes, Attributes and Relationships

Model Specification - Structure + Behavior

Structure - Classes + Relationships/Associations -> Information Model

- **Classes - entities in the application**
- **Attributes - Properties of classes**
- **Associations/Relationships among classes - “has_a”, “is_a,” etc.**
- **Constraints – Quantification of relationships among attributes, etc.**

Behavior - Transitions among states of entities -> State Model

- **States - Defined by assignments of values to attributes**
- **Transitions - Governed by state machines**
- **Events - Cause transitions.**

Analysis Process

1. Define Information Model
    1. Define classes and select attributes
    2. Define associations/relationships among classes
    3. Construct associative classes
    4. Iterate steps 1, 2 & 3 until consistent

2. Construct State Models
    1. Construct state model for each object with multiple states
    2. For each state model, define actions effecting state changes

3. Iterate until consistent

Development of Information Model

Methods for recognizing classes, attributes and relationships

1. **Linguistic analysis of requirements specification - textbook**
2. **Operational analysis of system behavior**
   - **Object Behavior Analysis – Rubin and Goldberg paper (related to Use Case analysis)**

Practical Approach - Apply both methods and compare resulting information model

Lecture – First carry through linguistic analysis and then sketch the object behavior analysis method in a later lecture.

Linguistic Analysis Method

1. Write Requirements Specification
2. Generate Class Diagram (except for state models) from Requirements Specifications
3. Construct Use Cases from Requirements
4. Generate State Model from Information Model and Use Cases

Relation to Other Knowledge Representation Processes

1. Software Engineering - Domain Analysis or Domain Engineering
2. AI - Conceptual Structures or Knowledge Networks,

Classes, Attributes and Relationships

Information Models via Semantic Analysis

Identify the classes . The classes derive from noun phrases. Go through the requirements statement and pick out all of the important nouns. Categorize these nouns into tangible items or devices, physical classes, roles, interactions, instance specifications, etc.. Decide which ones are significant and which ones are redundant and select classes.

Define attributes for each object. Attributes come from possessive phrases, as descriptions of the nouns. Recall that the attributes define the identity and the state of an object.

Define relationships. Relationships can be derived by looking at the verb phrases of the requirements analysis. Verb phrases include such things as the PC board "is made up of" chips and connectors, etc..

Specialize and generalize classes into subtypes and supertypes and create associative, create object definitions for m to n associations or dynamic associations which fall under the heading of events, interactions which have to be remembered.

Classes Are Akin To Pornography Or Beauty

tangible entity or device - airplane

role - professor, student

incident or persistent event - registration of automobile

interaction - contract terms

specification – sets of entities with related characteristics

organization - university

external systems - sensor drives

Attributes and Attribute Types

Naming - unique identifier for instances of object types.
May be single or multiple attributes
Descriptive - intrinsic properties of classes - color, size, etc.
Referential - Specifies relationships
- Identifies partners in has_a relationships
State or Status - Values define current state of object.
May be changed during lifetime of object instance.
Status - on/off, position of robot arm, etc.

Attribute Data Types

The set of values an attribute can take on constitutes its *data type.*

Each attribute must have a data type

1. Domain Specific and Domain Named
2. Instances of or structures of Core Data Types

Not basically different from constructed types for conventional programming systems

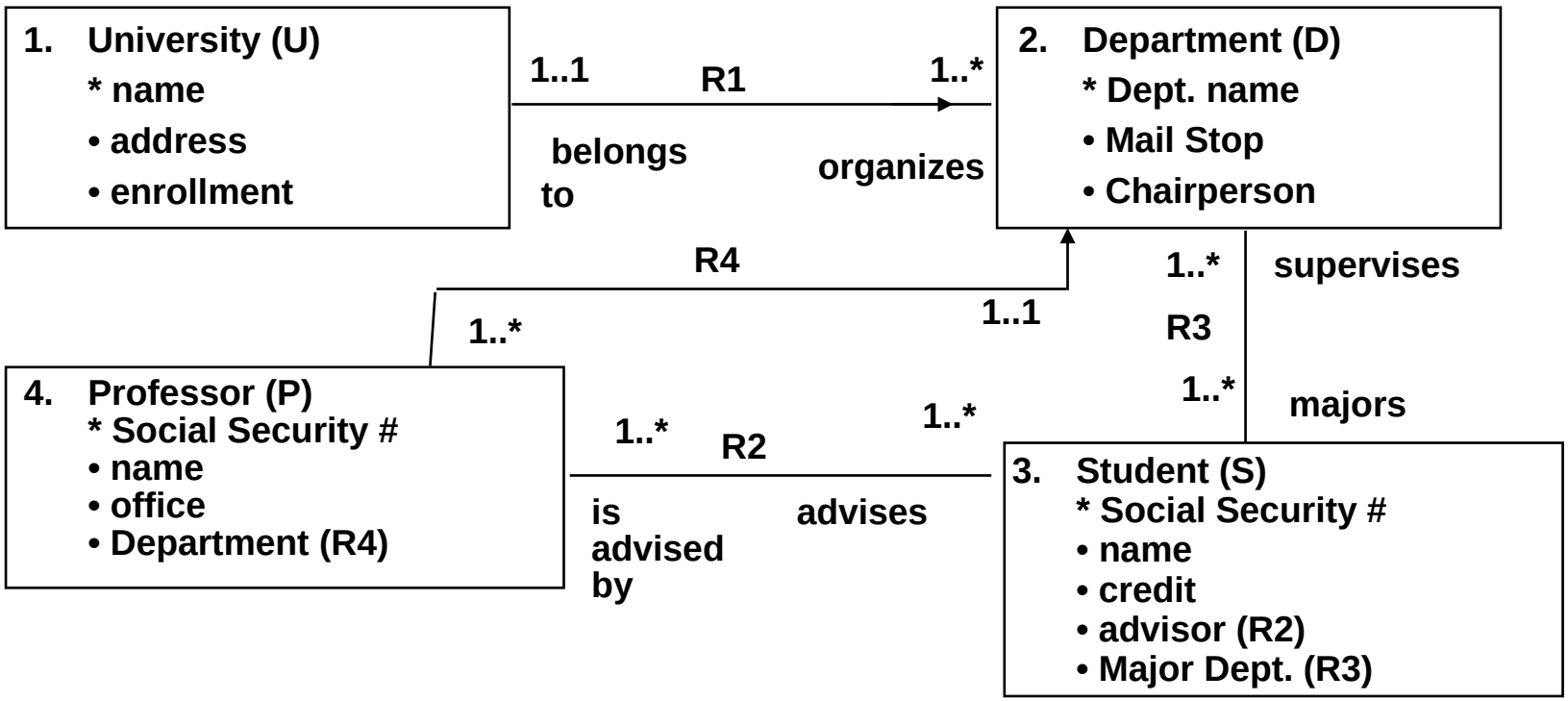
Core Types

***Definition:* A *core data type* is a fundamental datatype from which other domain-specific datatypes can be defined.**

Boolean, string, integer, real, date, timestamp, arbitrary_id

Classes, Attributes and Relationships

University where students have majors and advisors but advisors are independent of majors



Specifications for the Control Software for a One-Button Microwave

The microwave is powered by a powertube with a wattage rating and a part number, it incorporates: an interior light with a specified wattage and part number, a beeper which has a specified part number and a control button. The door of the oven is latched by a friction catch. In the current product the powertube consumes 600 watts and light consumes 40 watts.

This simple oven has a single control button. When the oven door is closed and the user presses the button, the oven will cook (that is, energize the power tube) for 1 minute.

There is a light inside the oven. Any time the oven is cooking, the light must be turned on, so you can peer through the window in the oven's door and see if your food is bubbling. Any time the door is open, the light must be on, so you can see your food or so you have enough light to clean the oven.

When the oven times out (cooks until the desired preset time), it turns off both the power tube and the light. It then emits a warning beep to signal that the food is ready.

The user can stop the cooking by opening the door. Once the door is opened, the timer resets to zero.

Closing the oven door turns out the light.

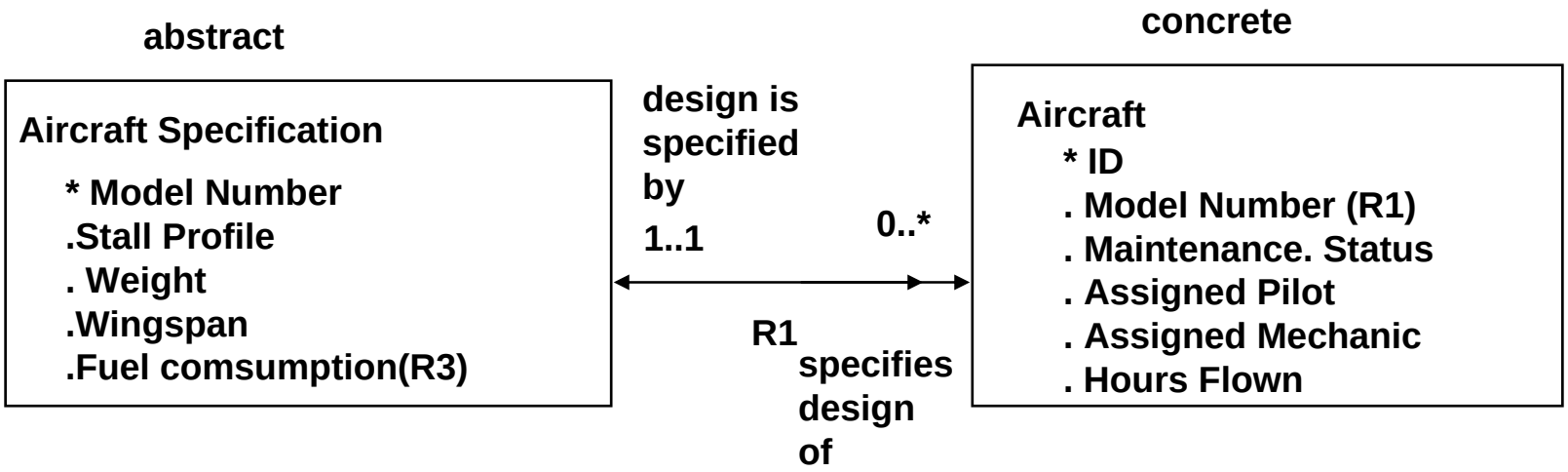
Specification Classes

An airplane leasing company owns many instances of a given airplane and leases several different types of airplanes.

The leasing company keeps track of the leasee and the maintenance operations of each plane. The data kept includes the pilot, the mechanic, operational status and number of hours flown for each airplane.

Classes, Attributes and Relationships

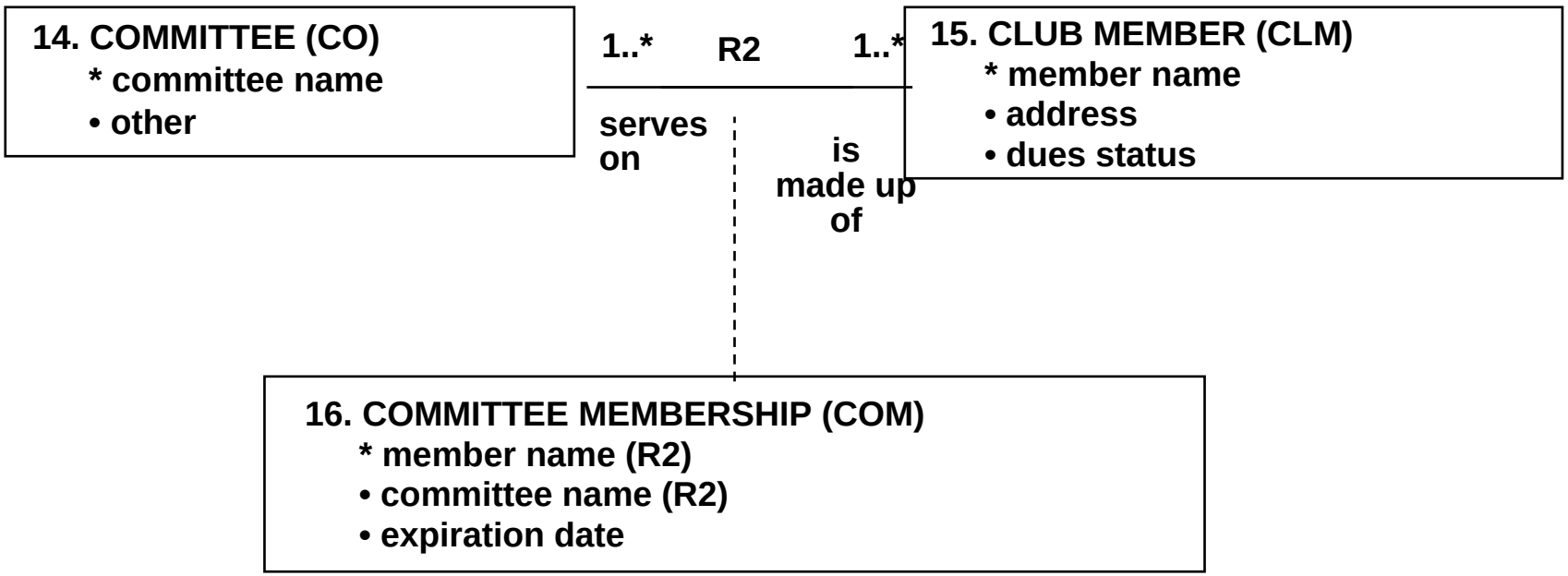
Specification classes



Why two classes rather than one?

Associative Classes and M to N Relationships

M ⟷ N relationships cannot be readily represented by referential attributes. M ⟷ N relationship ⟺ associative object

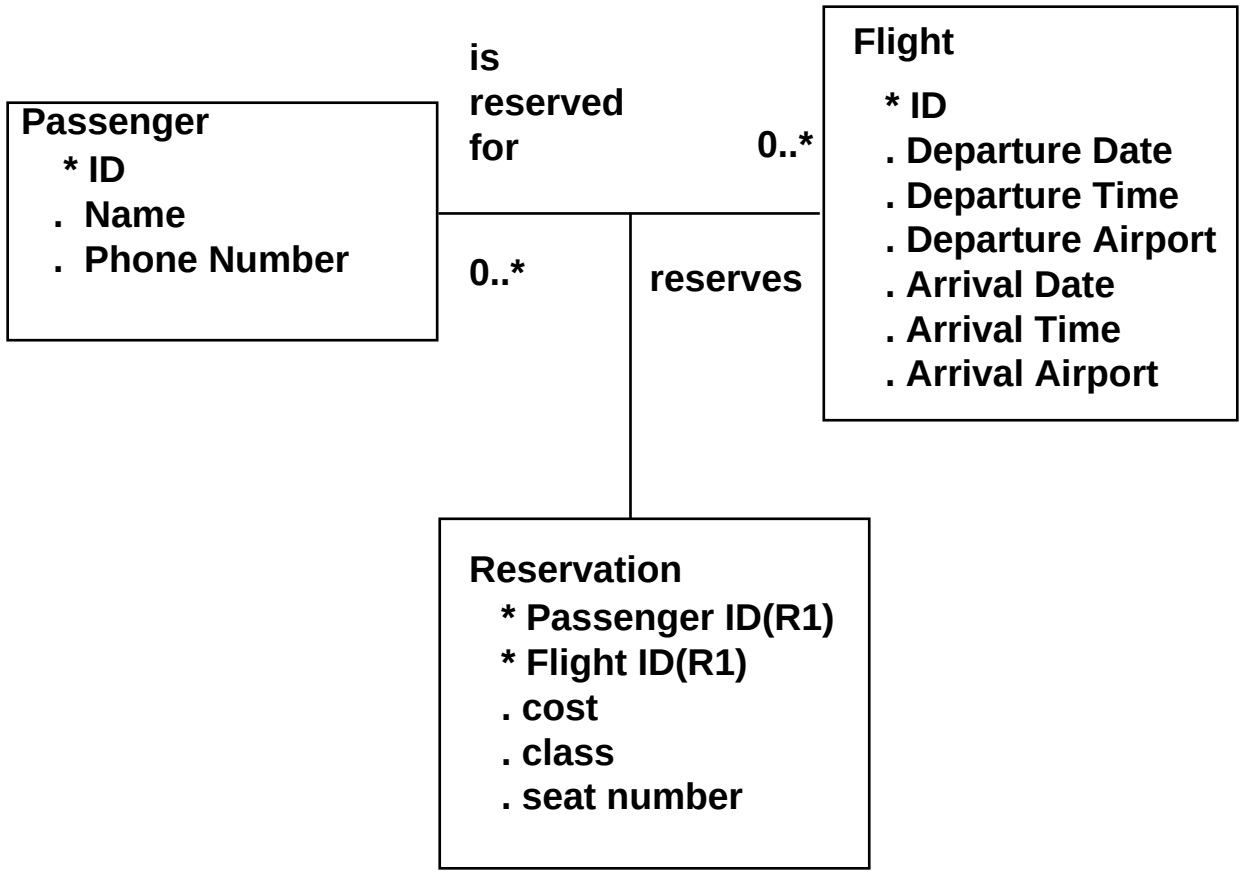


Interaction classes - Associative classes

An airlines reservation system sells reservations on flights and assigns seats on flights. The flights are characterized by departure airport, date and time and arrival airport, date and time. The passengers are characterized by name and telephone number. A reservation is for a passenger on a flight with a given seat at a specific cost. A passenger may book reservations on several flights and each flight has many passengers.

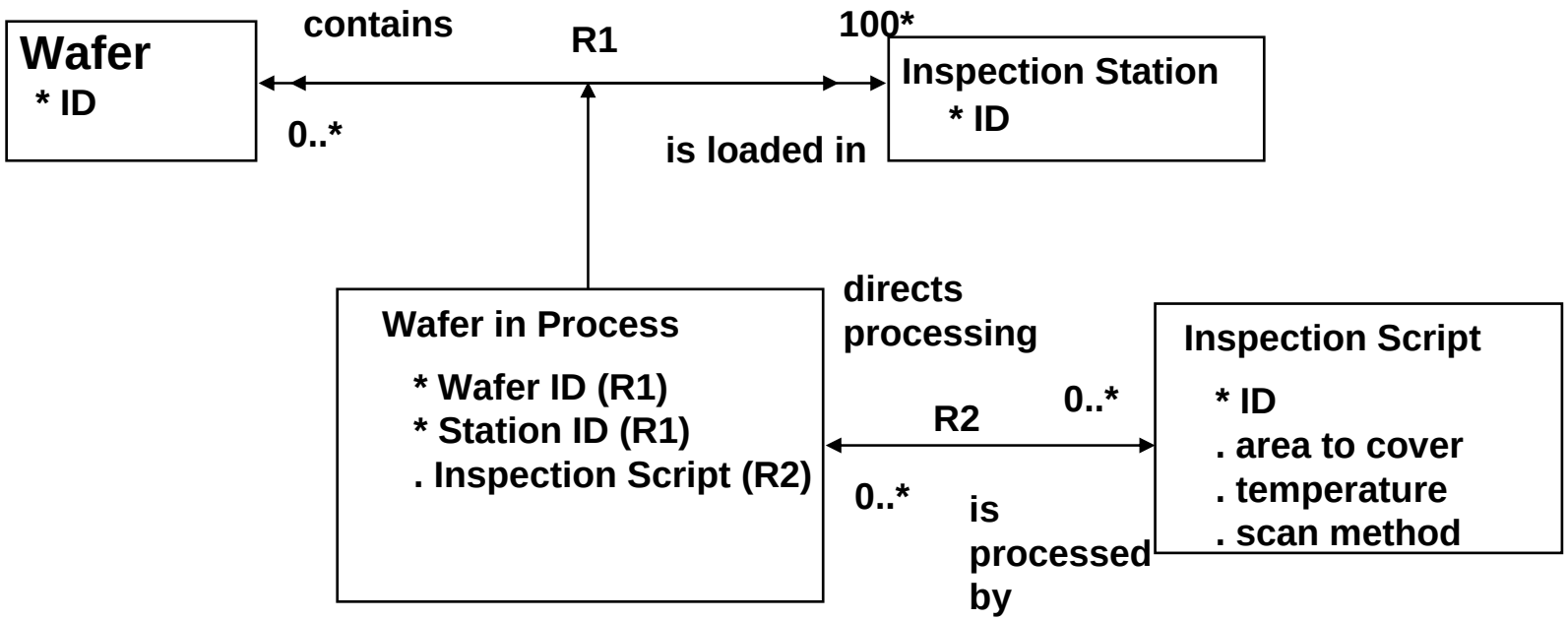
Classes, Attributes and Relationships

Interaction Object - Associative Object



Role classes as Associative classes

A silicon wafer is formed and then is inspected at an inspection station. Each wafer is processed through inspection stations by one or more inspection scripts. There are many wafers and several inspection stations. A wafer may undergo inspection under the control of several scripts. A wafer is regarded as "in process" until it completes all of its inspections.

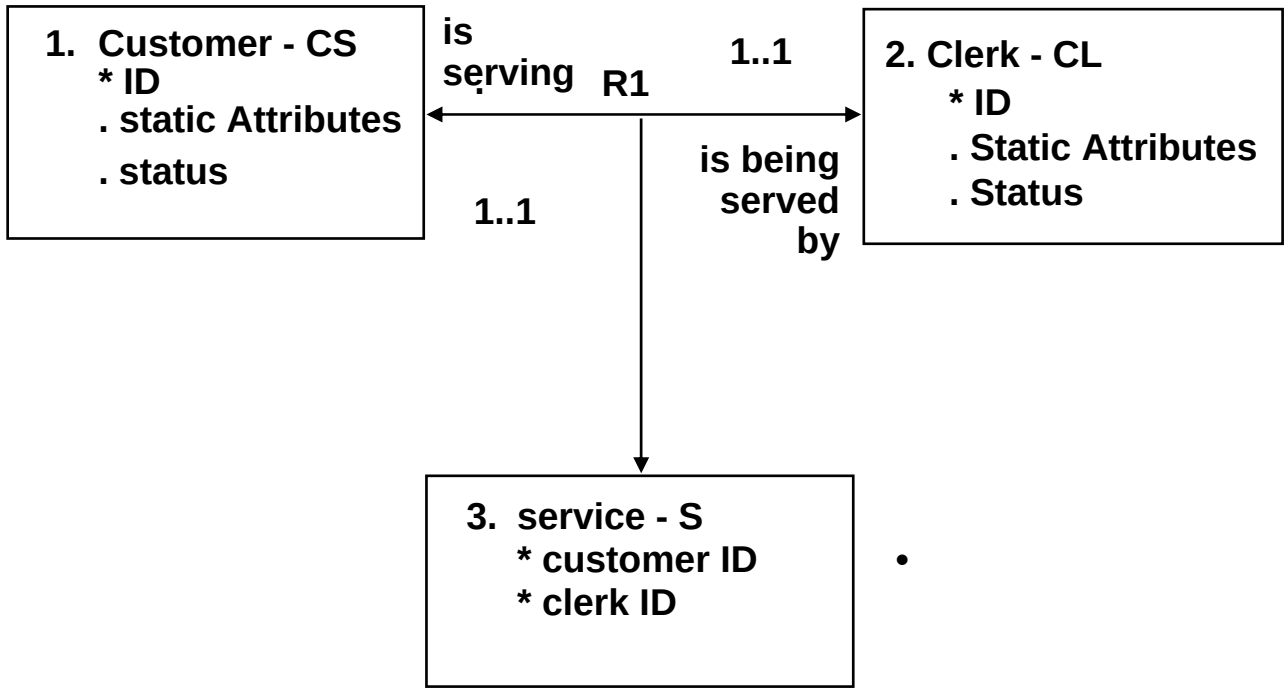


Competitive Relationships

Consider a service and commission management system for an upscale Women's Ware Store. The customers come into the store and wait for a clerk to show them merchandise. There are typically only a few clerks and at some times of day there are more customers than clerks.

This is an example of a competition for resources. It is representative of a commonly occurring circumstance.

Classes, Attributes and Relationships

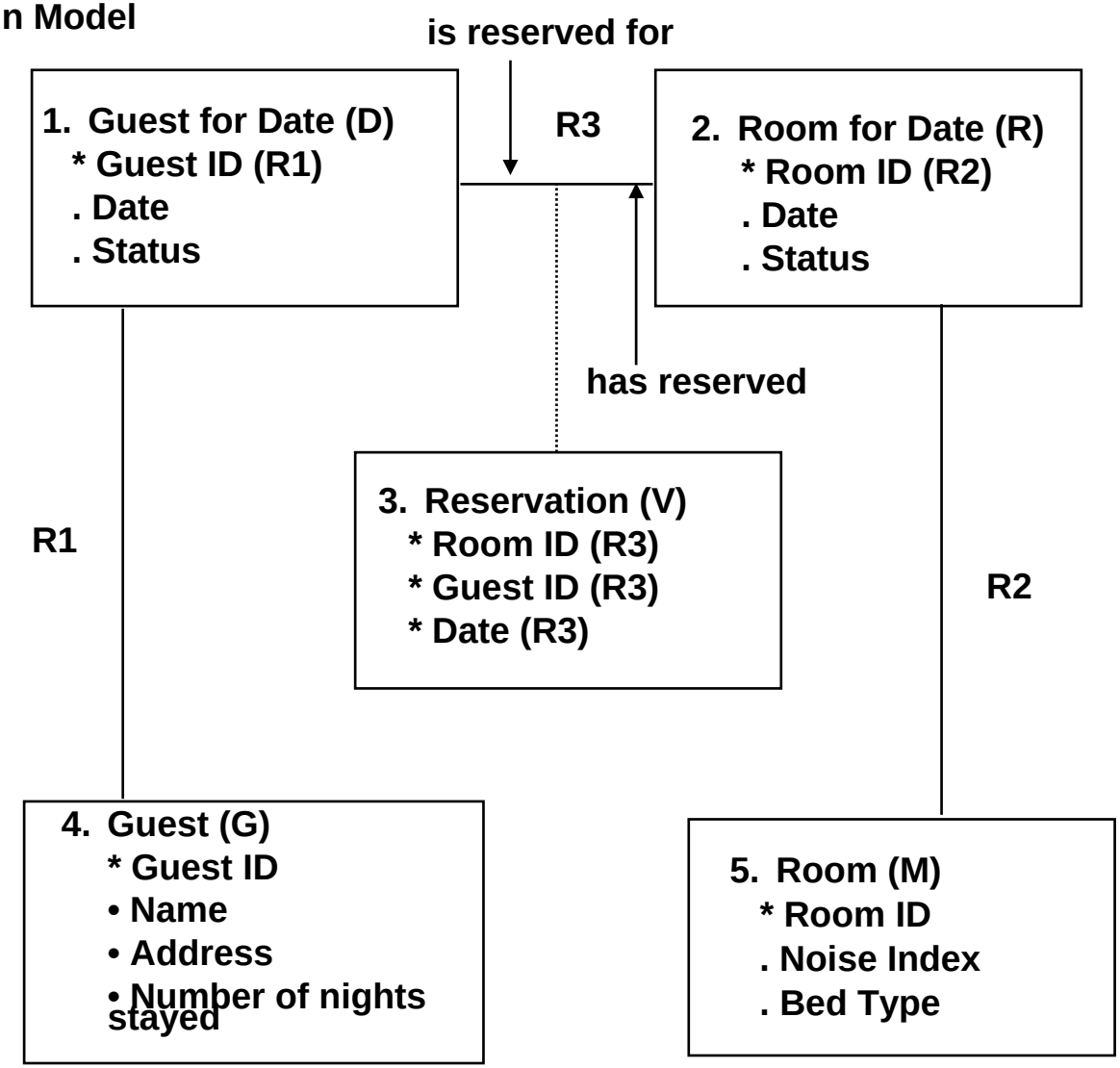


Requirements Statement for Hotel Room Reservation System

A hotel is developing a system for matching available rooms to requests for reservations from potential guests. The potential guest provides a request for a room for some dates and the system is to match the available supply of rooms available for specific dates to the requests for rooms on a given date.

Classes, Attributes and Relationships

Information Model



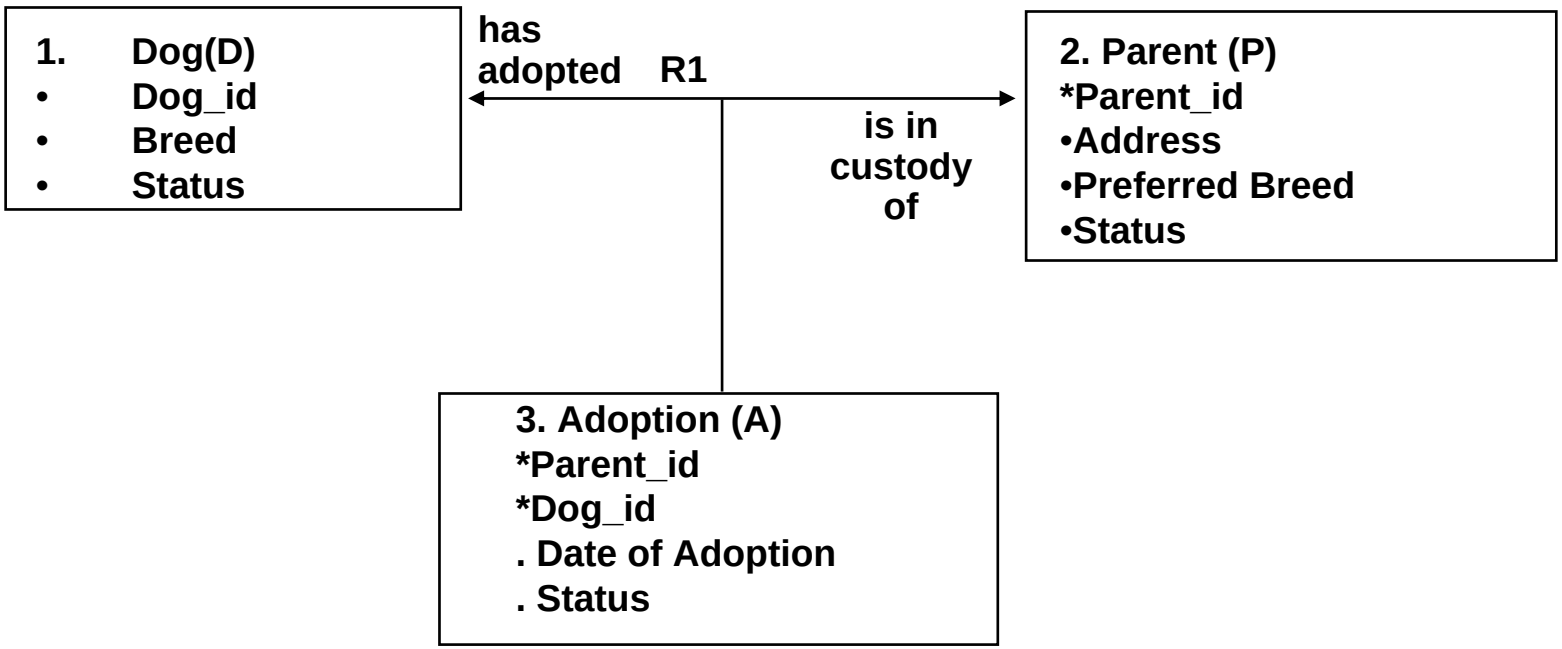
Dog Adoption Agency

A dog adoption agency accepts requests for dogs from potential adopters (=parents). The potential parent specifies the characteristics of the dog desired. Dogs are sent to the agency to be assigned to parents. When a parent makes an application then the agency tries to match the request. If a dog is available then an assignment is made. Note the possible competition for dogs considered desirable.

The parent also has the option of returning the dog after a six months probationary period and either asking for a new dog or withdrawing from the adoption program.

Competitive assigner where instance of Relationship Object has State Model

Information Model for Dog Adoption



Parent has 6 months to finalize adoption.

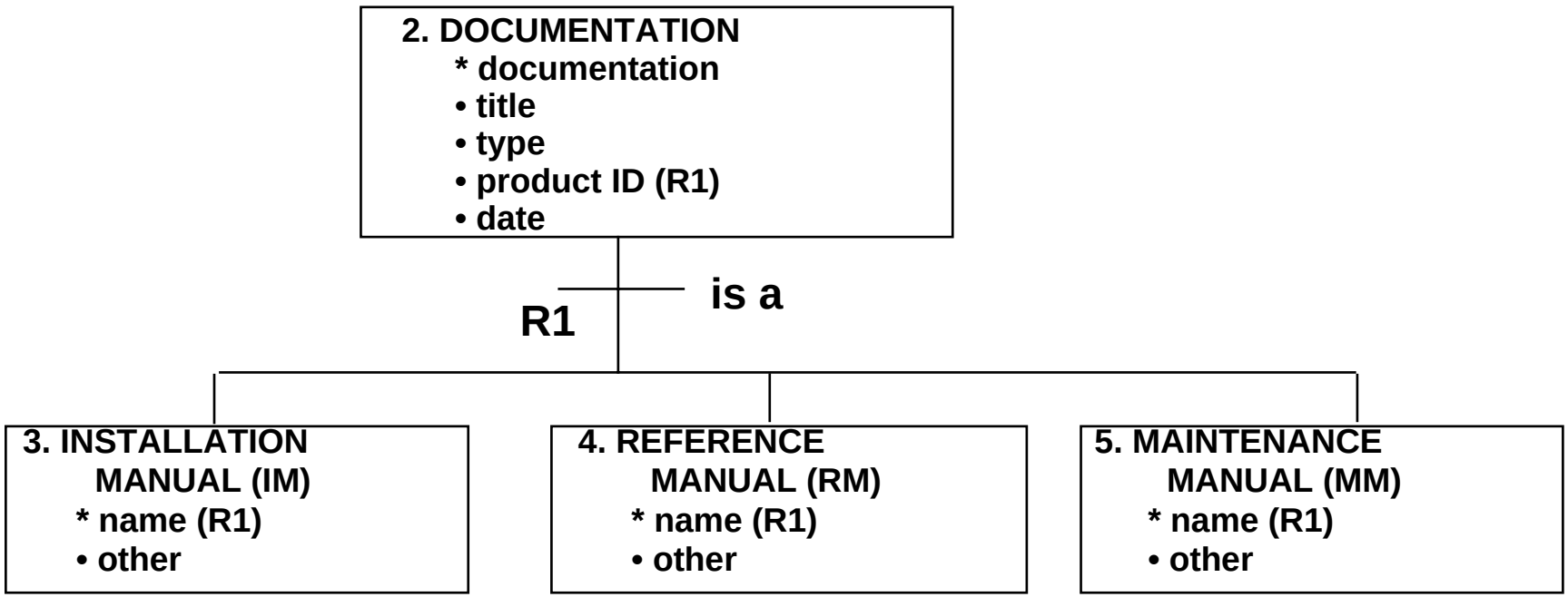
The Drug Manufacturing Problem

When a patent on a given prescription drug expires it becomes available for manufacture by companies other than the original patent holders. Potential manufacturers of the drug may apply for licenses to manufacture the drug. Each drug may require certain other drugs for its manufacture. To become licensed to manufacture a given drug requires that the manufacturer already hold a valid license for those drugs required for the manufacture of the drug for which the application is made or have an application for a license to all of the constituent drugs. A given manufacturer cannot simultaneously hold licenses for more than k drugs. If there are simultaneous applications by a single manufacturer or licenses for a given drug and its constituent drugs either all are approved or all are rejected. The applications are evaluated in FIFO order and those applicants who qualify are licensed to manufacture the drug for some fixed duration. Applications which do not qualify are rejected and deleted from the system.

"is a" Relationships

Specialization of classes to serve more differentiated roles.

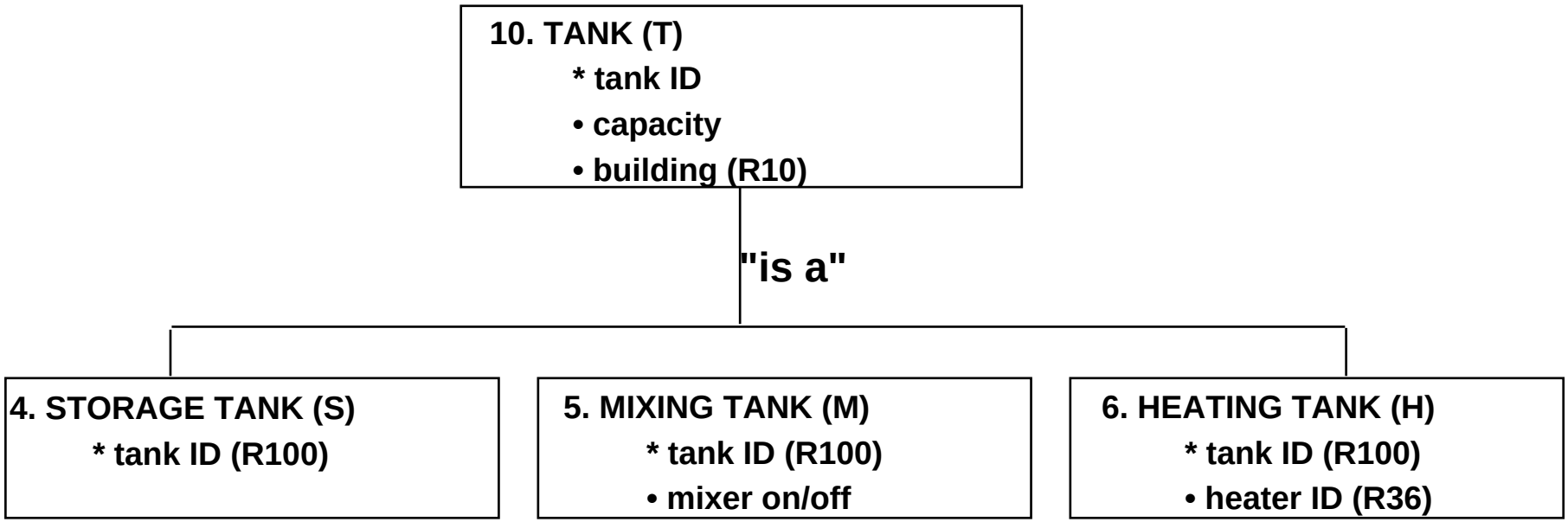
Similar to "inheritance" in object-oriented programming languages.



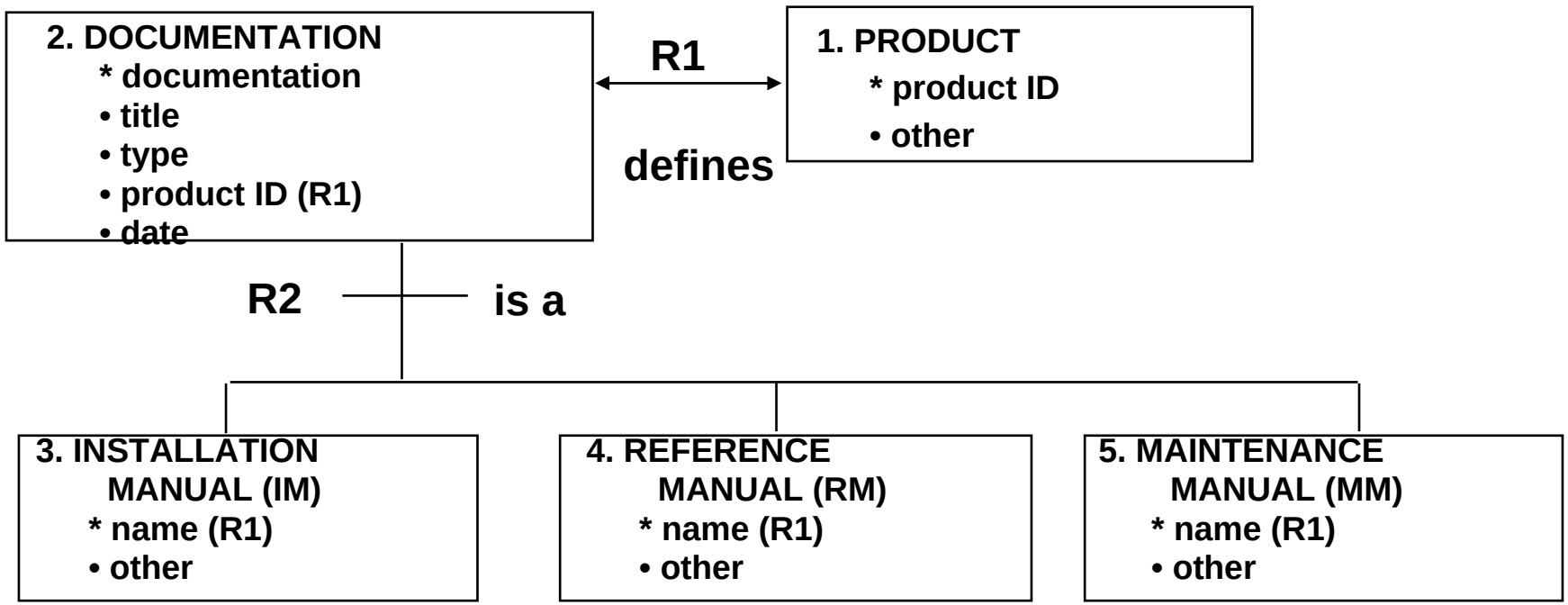
Classes, Attributes and Relationships

Examples of classes/Relationships (Shlaer/Mellor)

Juice Factory Process Control System

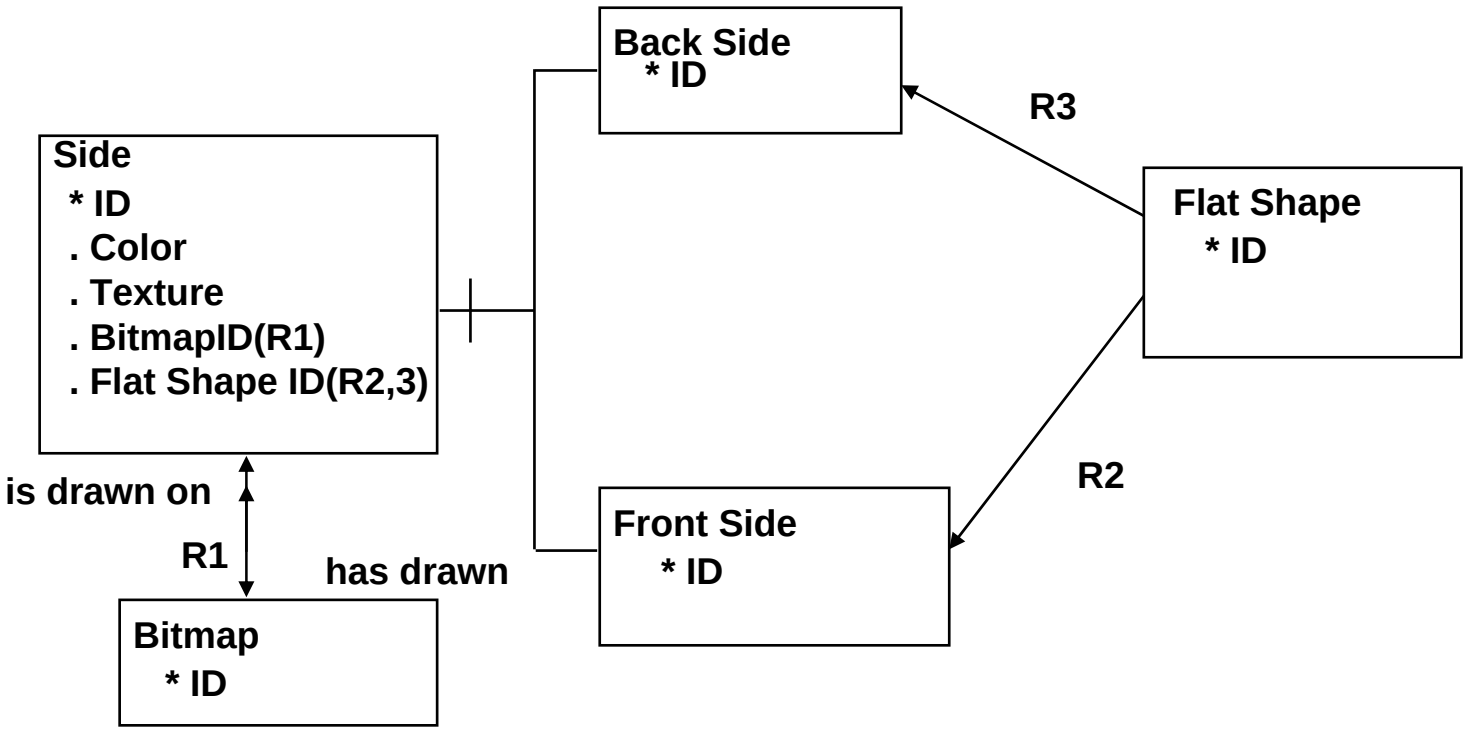


EXAMPLE OF "has" AND "is a" RELATIONSHIPS



Role classes

A flat object has a front side and a back side. An object model for a circumstance where it is needed to include front and back sides can be represented as follows.



Mixing Tank Specification

A batch of product is mixed in a mixing tank and then transferred to a bottling tank for bottling. The two subtypes of tanks have different state machines.

The bottling tank is cleaned and prepared while a batch of product is in the mixing tank. The time taken to clean and prepare the bottling tank for a new batch is variable. Notification that the bottling tank is ready may come before or after mixing is completed. The software system must be constructed to deal with both circumstances.

Classes, Attributes and Relationships

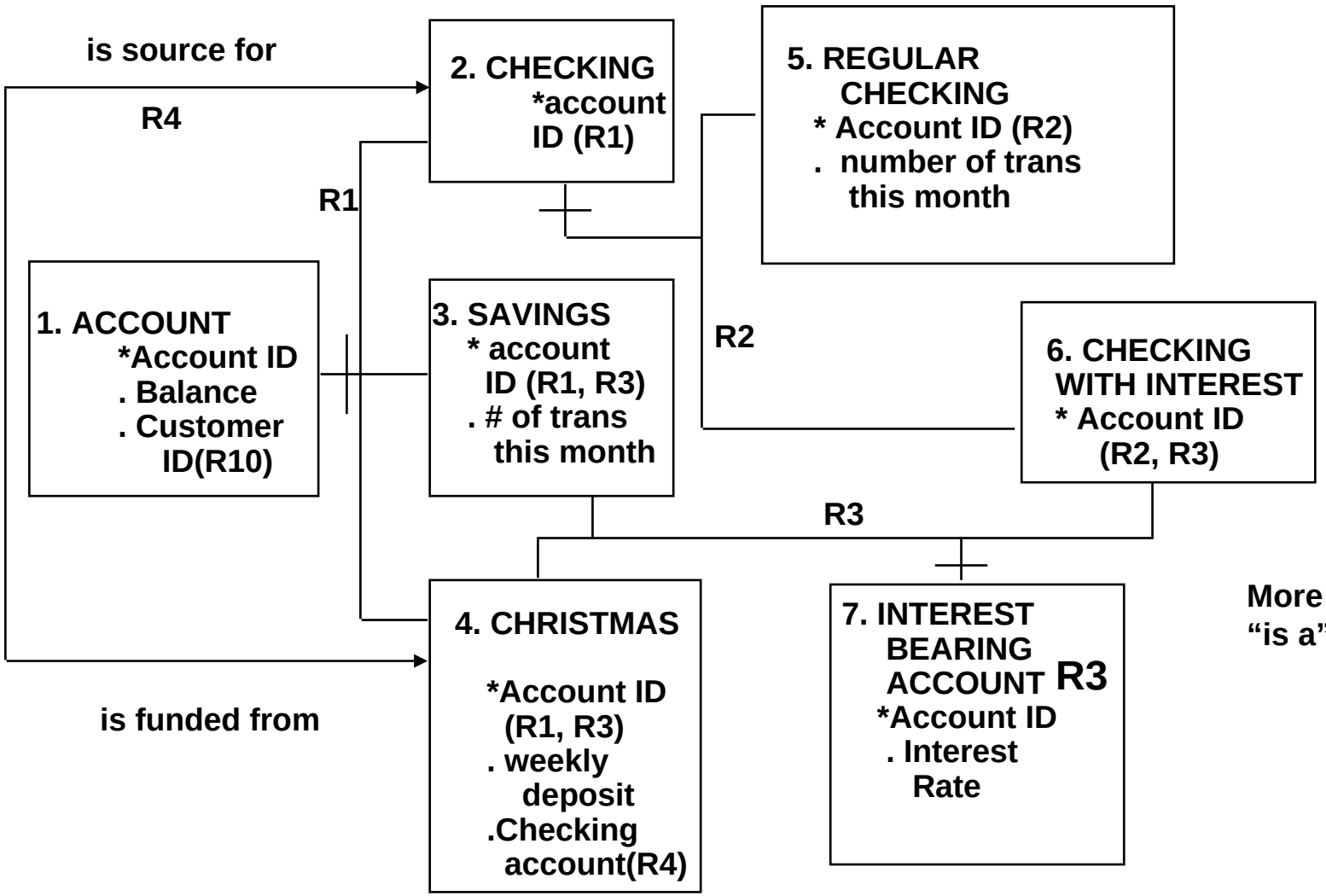
Requirements Analysis for Checking Accounts

There are three types of accounts: checking accounts, savings accounts and automatically-funded accounts or "Christmas Club" accounts. Some types of accounts bear interest and some types of accounts do not bear interest. For example, a regular checking account does not bear interest. On the other hand, a savings account, a Christmas Club account or a checking account with interest are all interest-bearing accounts.

All accounts have an account ID, a balance and a customer ID. A checking account has fees determined by the number of transactions each month. A savings account has a limit on the number of transactions in a given month. A Christmas Club account is tied to a checking account from which a weekly deposit amount is transferred. An interest-bearing checking account has an interest rate which it pays as does a savings account and a Christmas Club account. An interest-bearing checking account also has a limit on the number of transactions per month.

The task is to construct an information model reflecting the "has a" and "is a" relationships of this requirements analysis.

Classes, Attributes and Relationships



More complex
"is a" relationships.