Object Modeling

CSCE 740 - Lecture 16 - 10/24/2017

Objectives for Today

- Introduce methods for starting an object model.
    - Identifying classes, their attributes, and their operations.
    - Identifying associations between classes.
- Get some experience with OO design.

An Approach for Object Modeling

- Start with a problem statement.
  - High-level requirements
- Identify potential objects.
  - Look for nouns.

A library has books, videos, and CDs that it loans to its users. All library material has an ID number and a title. Book 101.1 is The Wee Free Men by Terry Prachett.

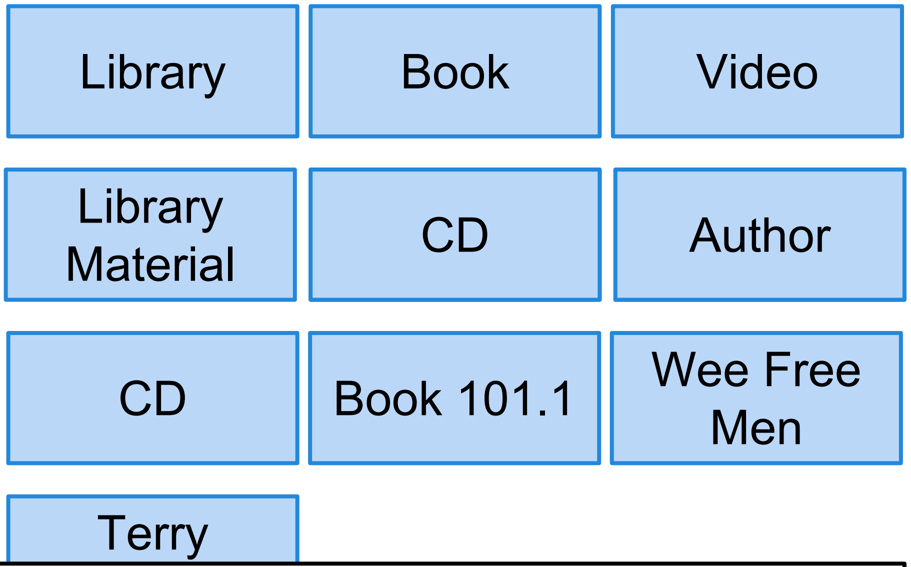
Library | Book | Video

CD | Library Material | ID Number | Title

Book 101.1 | Wee Free Men | Terry Prachett

Object Modeling Approach

- Refine and remove bad classes
  - Redundant, vague, or irrelevant.
  - Abstract objects to classes.
- Prepare data dictionary
  - Describe each class and its purpose.

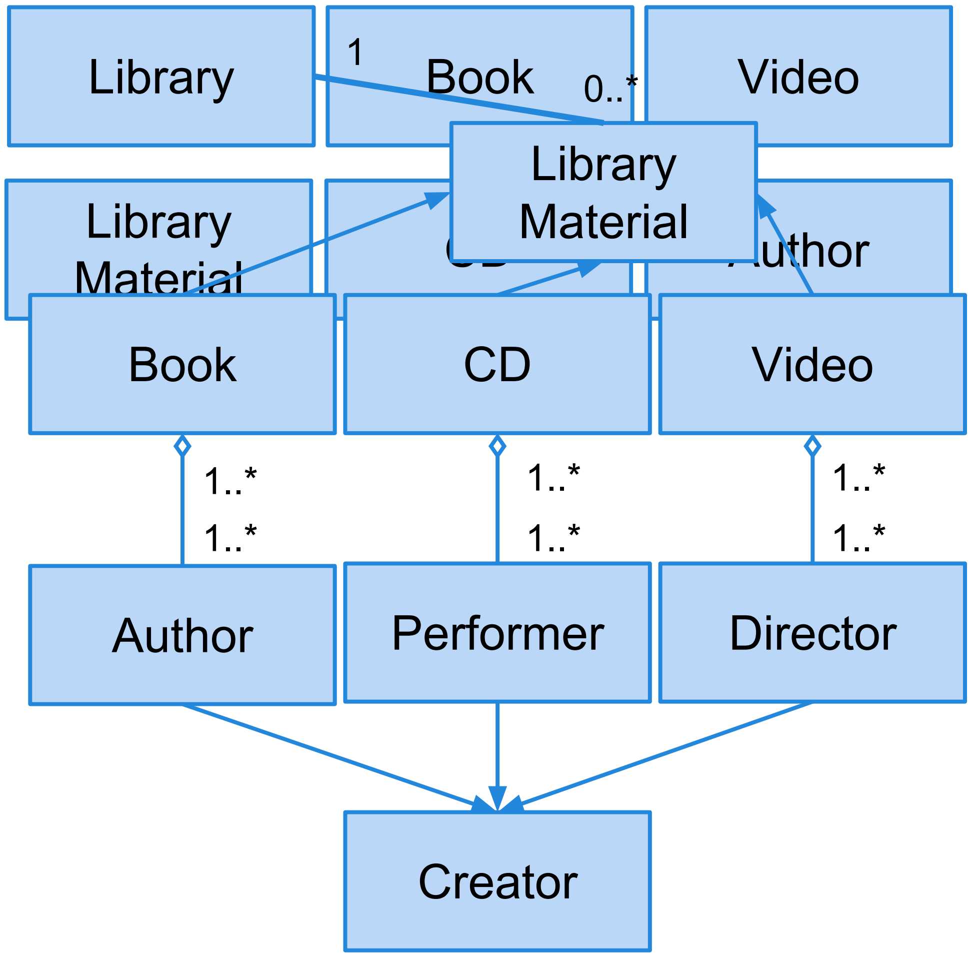


Library Material: An abstract class representing a generic library item that can be checked out. Has an ID and title.

Book: A class representing a book that can be checked out. Has an author in addition to inherited attributes ID and title.

Object Modeling Approach

- Identify associations and aggregations.
- Identify the attributes and operations of classes.
- Organize and simplify using inheritance.



Define Attributes and Operations

- What are the *responsibilities* of the class?
  - Use tools such as data dictionaries to define responsibilities of a class - what services must they perform or allow others to perform.
  - Classes were nouns, now look for **verbs**.
- General guidelines:
  - Responsibilities should be evenly distributed between classes.
  - Information related to a responsibility should be stored in the class responsible for that service.
    - Those are the attributes.

Identify Associations

Classes fulfill responsibilities in two ways:

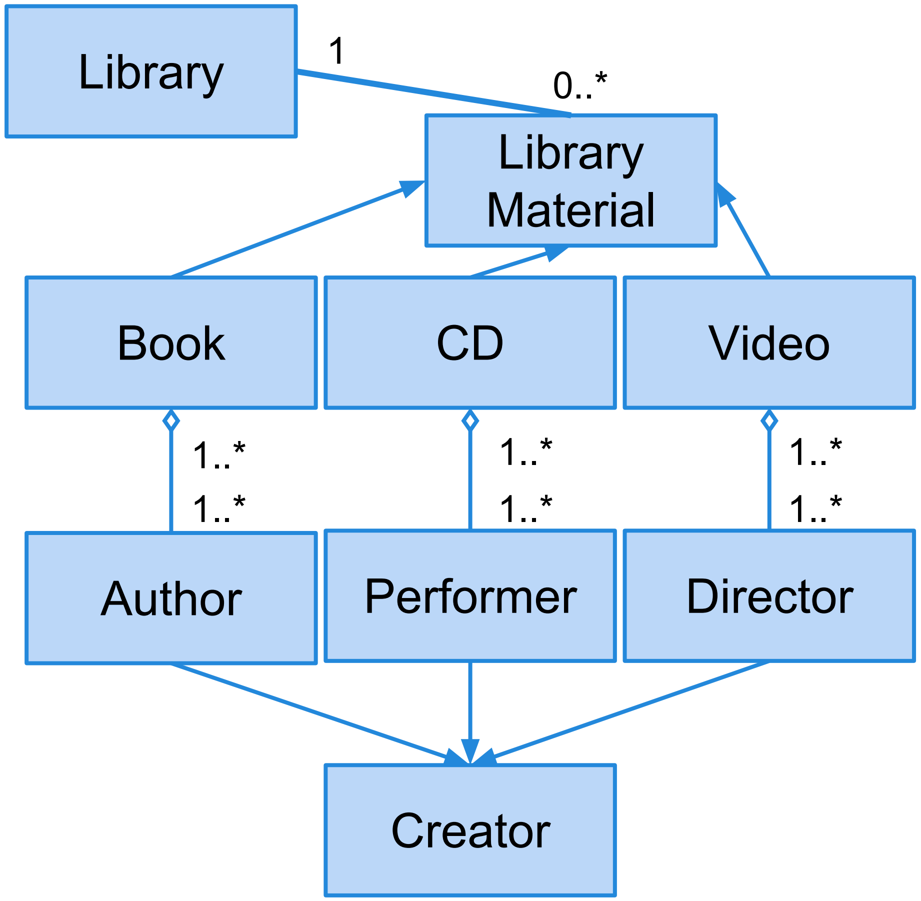
- It can use its own methods to modify its own attributes.
- It can collaborate with other classes.

If a class cannot fulfill its responsibilities alone, identify and document the associations.

- is-part-of (aggregation)
- has-knowledge-of (association)
- depends-upon (association)

Object Modeling Approach

- Iterate and refine the model.
  - You will almost always go through multiple iterations of a design.
- Group classes into subsystems.
  - Which classes can combine to form an independent grouping?



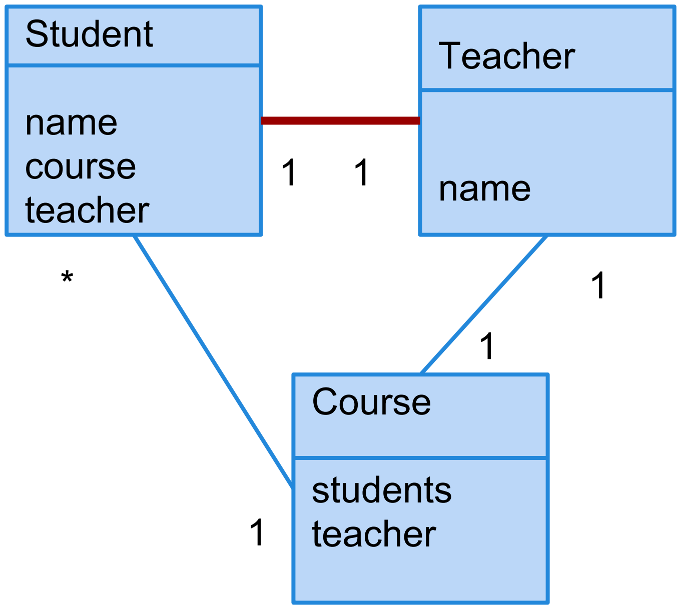
Refinement

The software design is often not optimal. Before implementation, consider how it can be improved.

Watch for:

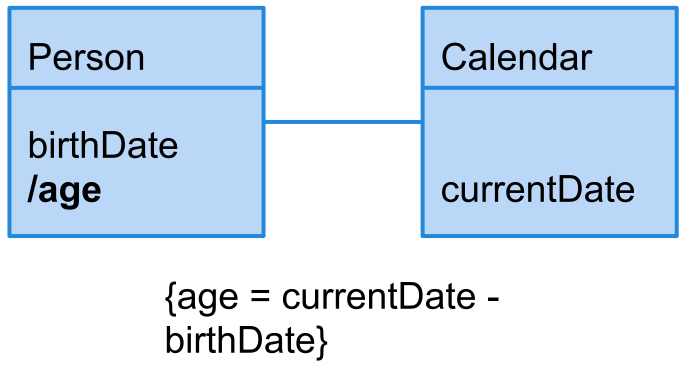
- Redundant associations.
- Attributes that can be derived at runtime.

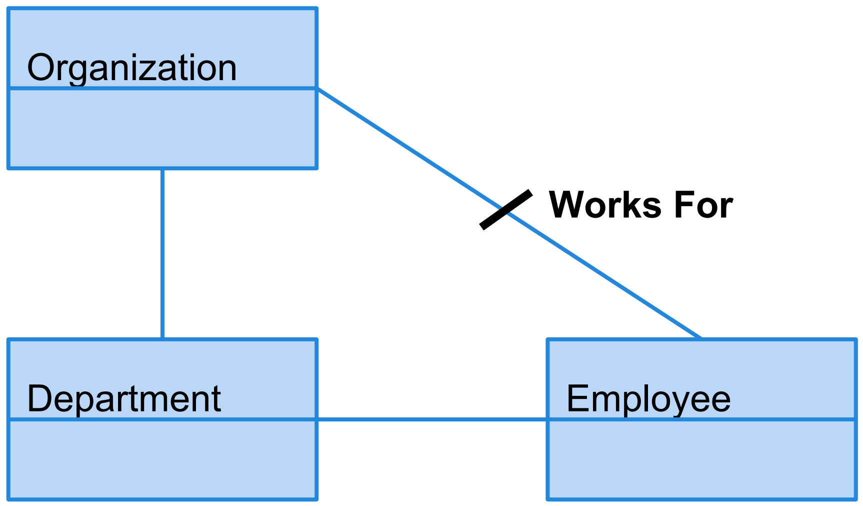
- Remove redundant associations



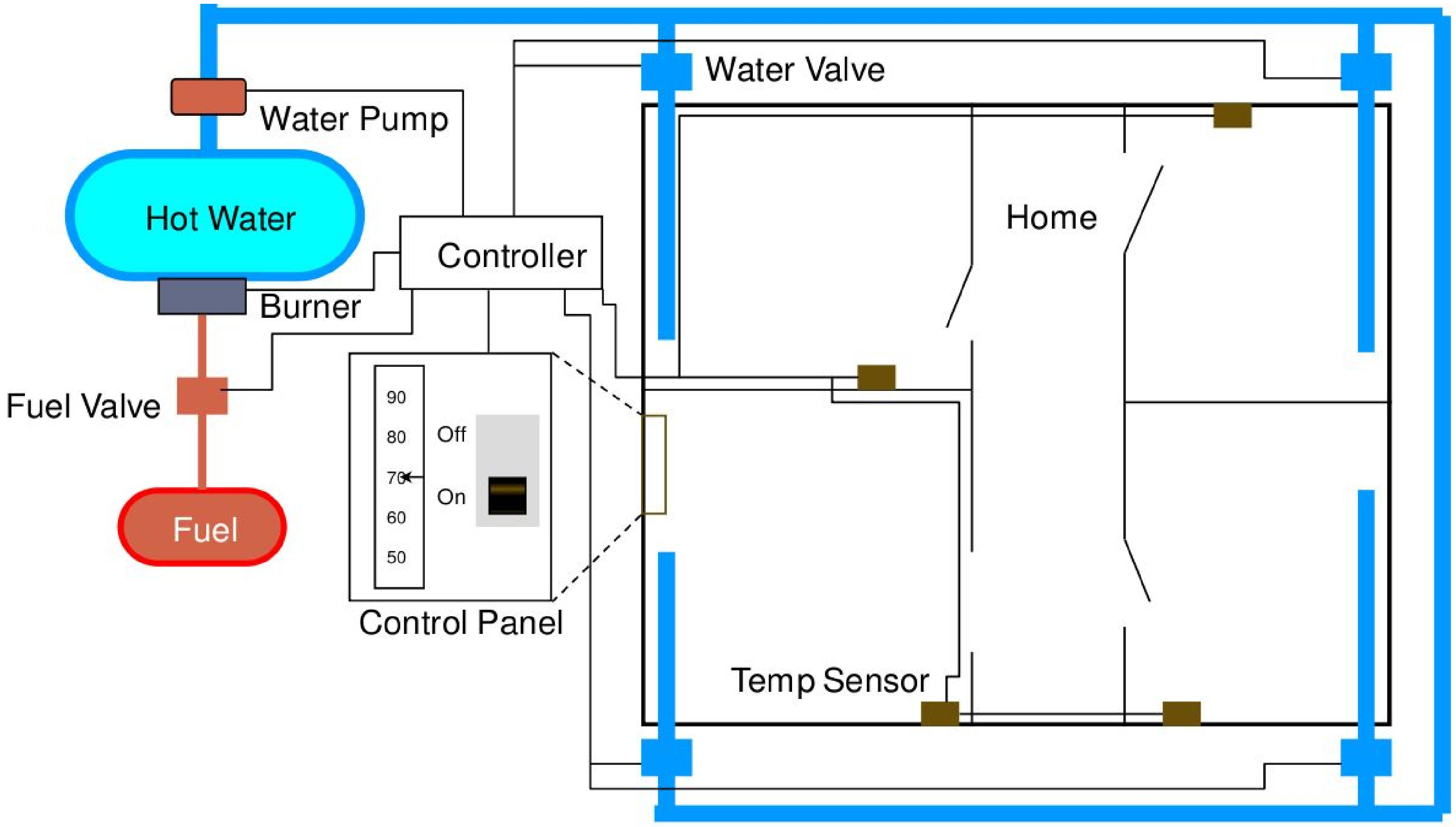
Derived Links and Attributes

Derived entities can be calculated from other entities. Indicated by a slash. They are potentially redundant.





The Home Heating System



Home Heating Requirements

The purpose of the software for the Home Heating System is to control the heating system that heats the rooms of a house. The software shall maintain the temperature of each room within a specified range by controlling the heat flow to individual rooms.

- The software shall control the heat in each room
- The room shall be heated when the temperature is 2F below desired temp
- The room shall no longer be heated when the temperature is 2F above desired temp
- The flow of heat to each room shall be individually controlled by opening and closing its water valve
- The valve shall be open when the room needs heat and closed otherwise
- The user shall set the desired temperature on the thermostat
- The operator shall be able to turn the heating system on and off
- The furnace must not run when the system is off
- When the furnace is not running and a room needs heat, the software shall turn the furnace on
- To turn the furnace on the software shall follow these steps
    - open the fuel valve
    - turn the burner on
- The software shall turn the furnace off when heat is no longer needed in any room
- To turn the furnace off the software shall follow these steps
    - close fuel valve
    - turn burner off

Identify Object Classes

$$\begin{tikzcd}
\text{Requirements Statements} \arrow[r] & \boxed{\text{Extract Nouns}} \arrow[r] & \text{Tentative Classes} \arrow[r] & \boxed{\text{Eliminate Classes}} \arrow[r] & \text{Final Class List}
\end{tikzcd}$$

Water Pump
Hot Water
Burner
Furnace
Fuel Valve
Fuel
Desired Temperature
On-Off Switch
Heating System

House
Room
Temperature
Home
Thermostat
Range
Control Panel
Heat Flow
Home Heating System

Water Valve
Controller
Software
User
Heat
Operator

Eliminate Bad Classes

- Redundant Classes
  - Classes that represent the same thing with different words.
- Irrelevant Classes
  - Classes we simply do not care about.
- Vague Classes
  - Classes with ill-defined boundaries.
- Attributes
  - Things that describe or make up other classes.

Eliminate Bad Classes (Continued)

- Operations
  - Sequences of actions are often mistaken for classes.
- Roles
  - The name of a class should reflect what it is, not the role it plays.
- Implementation Details
  - Save those for the implementation.

Identify Object Classes

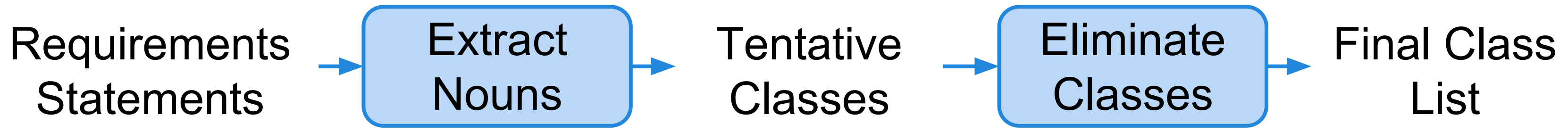
$$\begin{tikzcd}
\text{Requirements Statements} \arrow[r] & \boxed{\text{Extract Nouns}} \arrow[r] & \text{Tentative Classes} \arrow[r] & \boxed{\text{Eliminate Classes}} \arrow[r] & \text{Final Class List}
\end{tikzcd}$$

Water Pump
~~Hot Water~~
Burner
Furnace
Fuel Valve
~~Fuel~~
~~Desired Temperature~~
On-Off Switch
~~Heating System~~

~~House~~
Room
~~Temperature~~
~~Time~~
Thermostat
~~Range~~
Control Panel
~~Heat Flow~~
Home Heating System

Water Valve
Controller
~~Software~~
~~User~~
~~Heat~~
Operator

Classes After Elimination



Water Pump
Room
Burner
Furnace
Fuel Valve
Operator
Control Panel
On-Off Switch
Home Heating System

Water Valve
Controller
Thermostat

Prepare Data Dictionary

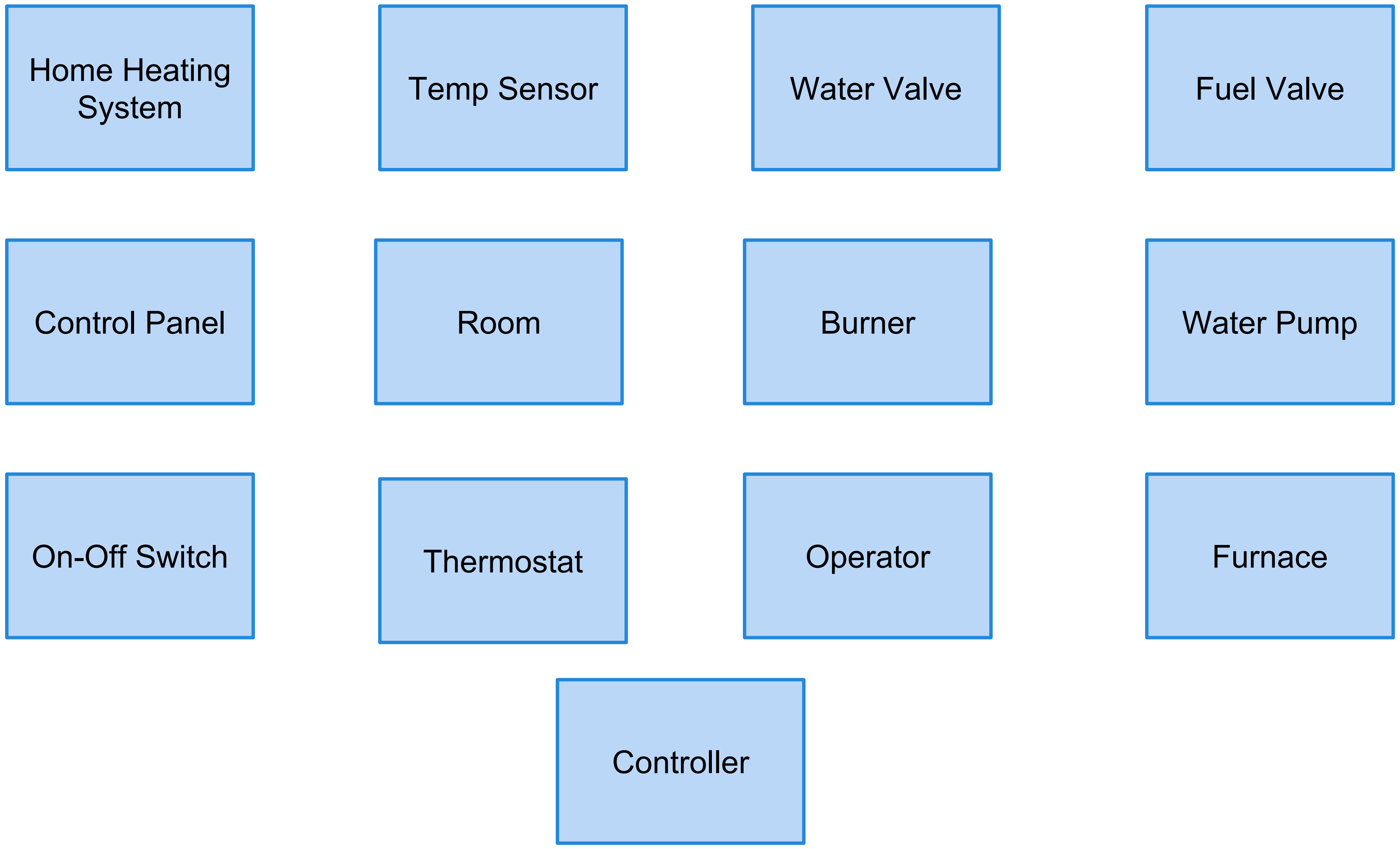
- Describe each class and its purpose.
- What are the classes' responsibilities? What information does it need to perform those services?

Derive Possible Associations

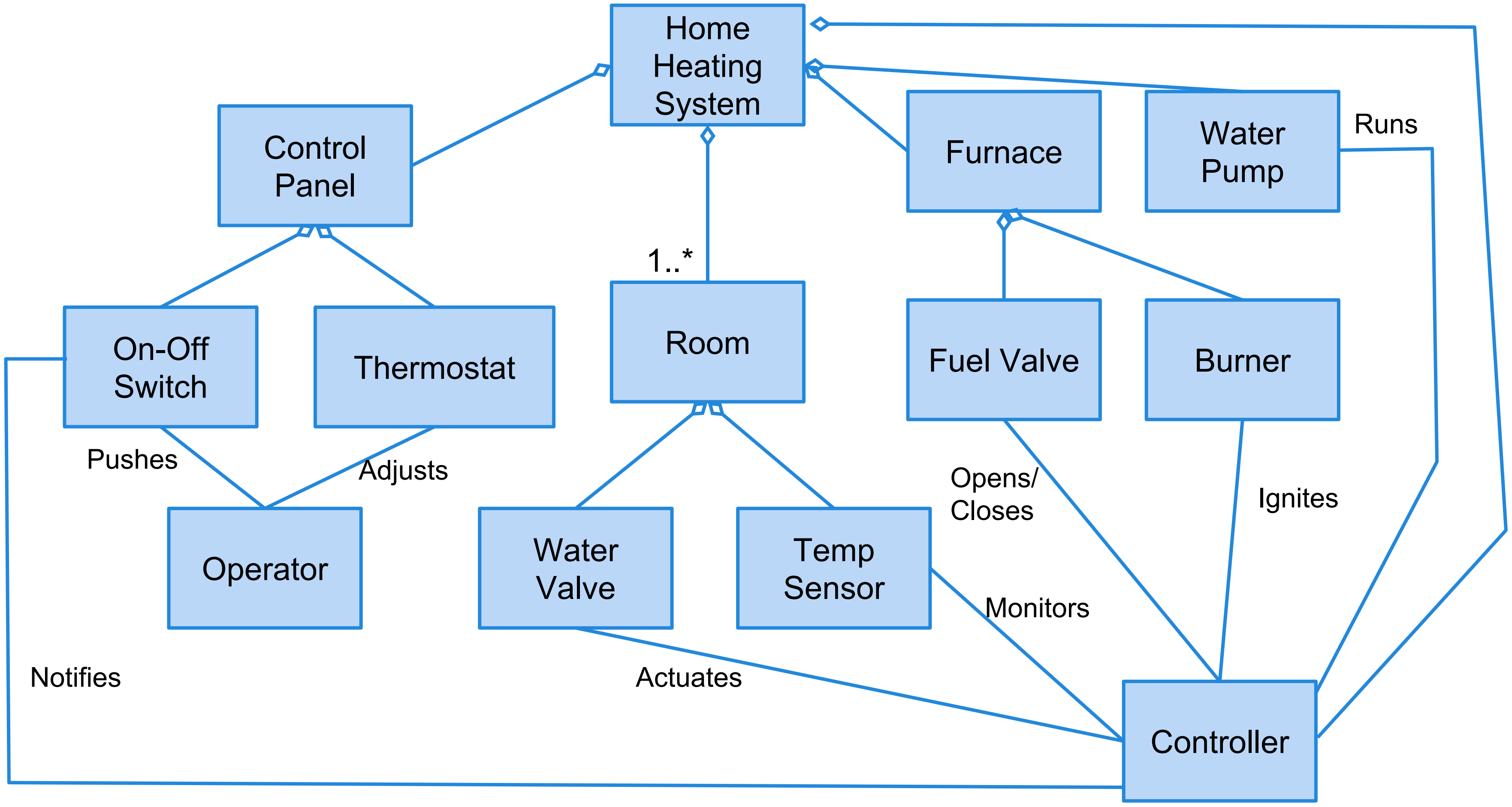
- Not much information from the written requirements.
- … but, a lot of information from the data dictionary and physical design.

- A room consists of a thermometer and a radiator
- A radiator consists of a valve and a radiator element
- The home heating system consists of a furnace, rooms, a water pump, a control panel, and a controller
- The furnace consists of a fuel pump and a burner
- The control panel consists of an on-off switch and a thermostat
- The controller controls the fuel pump, the burner, and the water pump. It monitors the temperature in each room, and opens and closes the valves in the rooms
- The operator sets the desired temperature, and turns the system on and off
- The controller gets notified of the new desired temperature

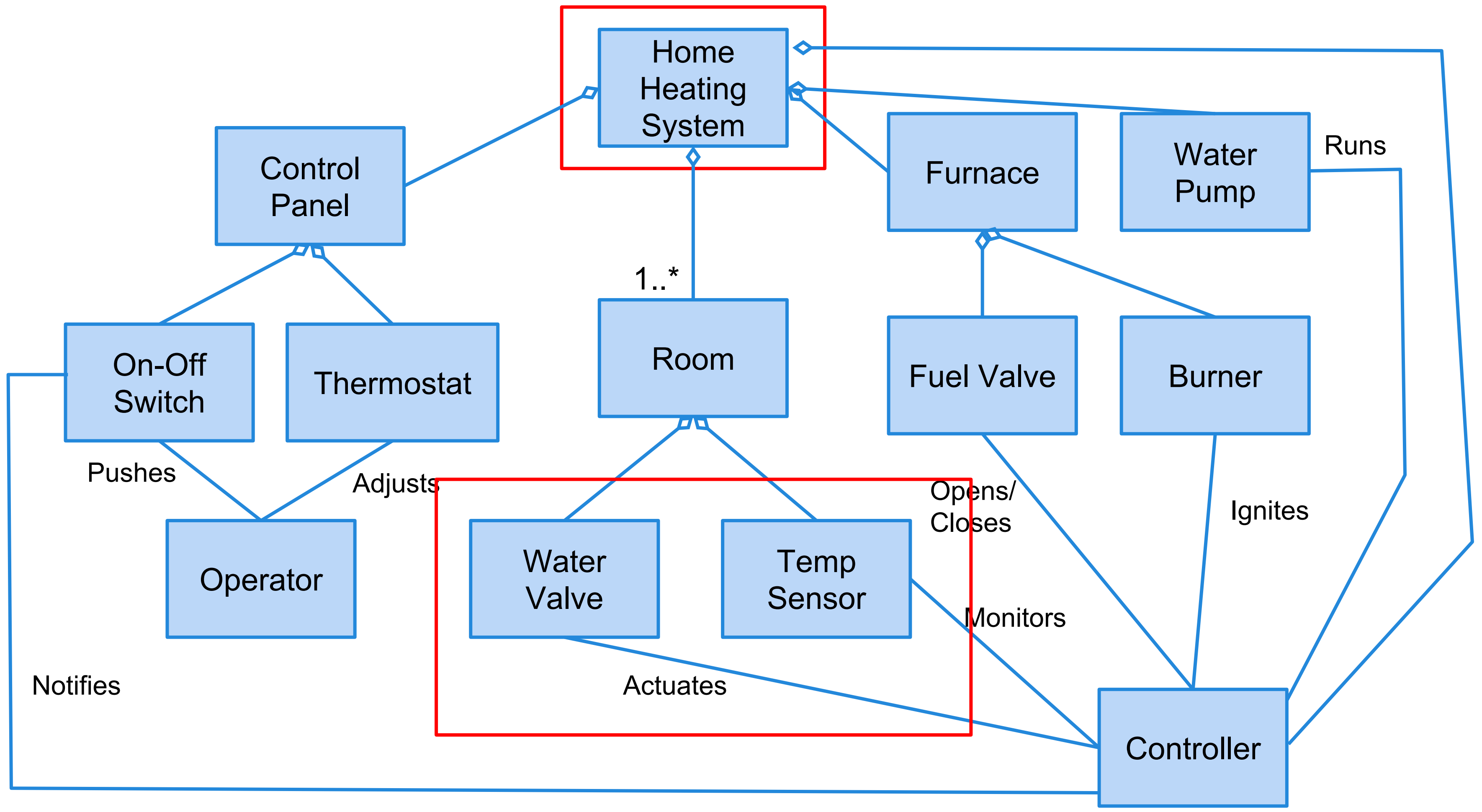
Add Associations to Complete the Class Diagram



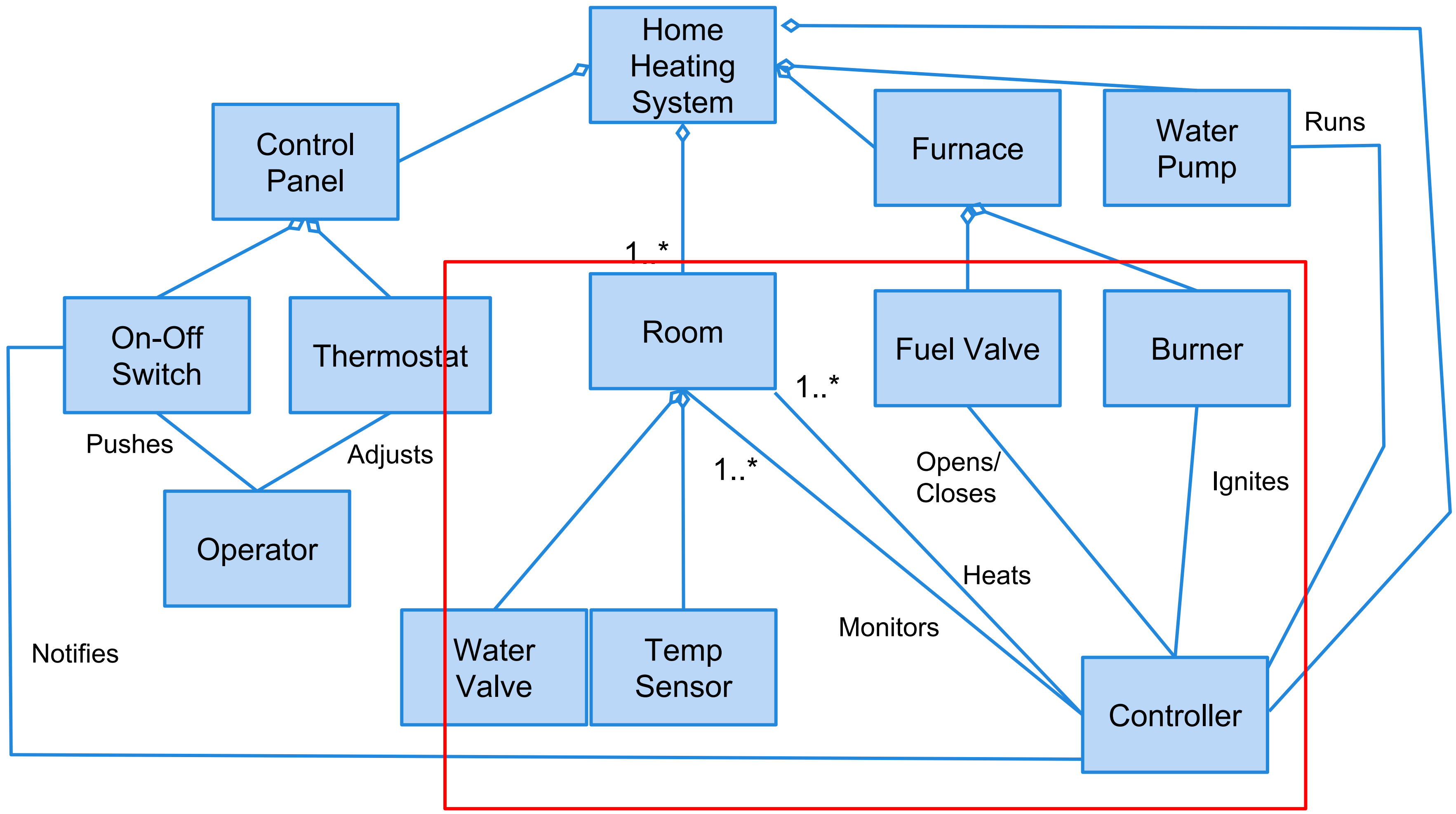
Class Diagram



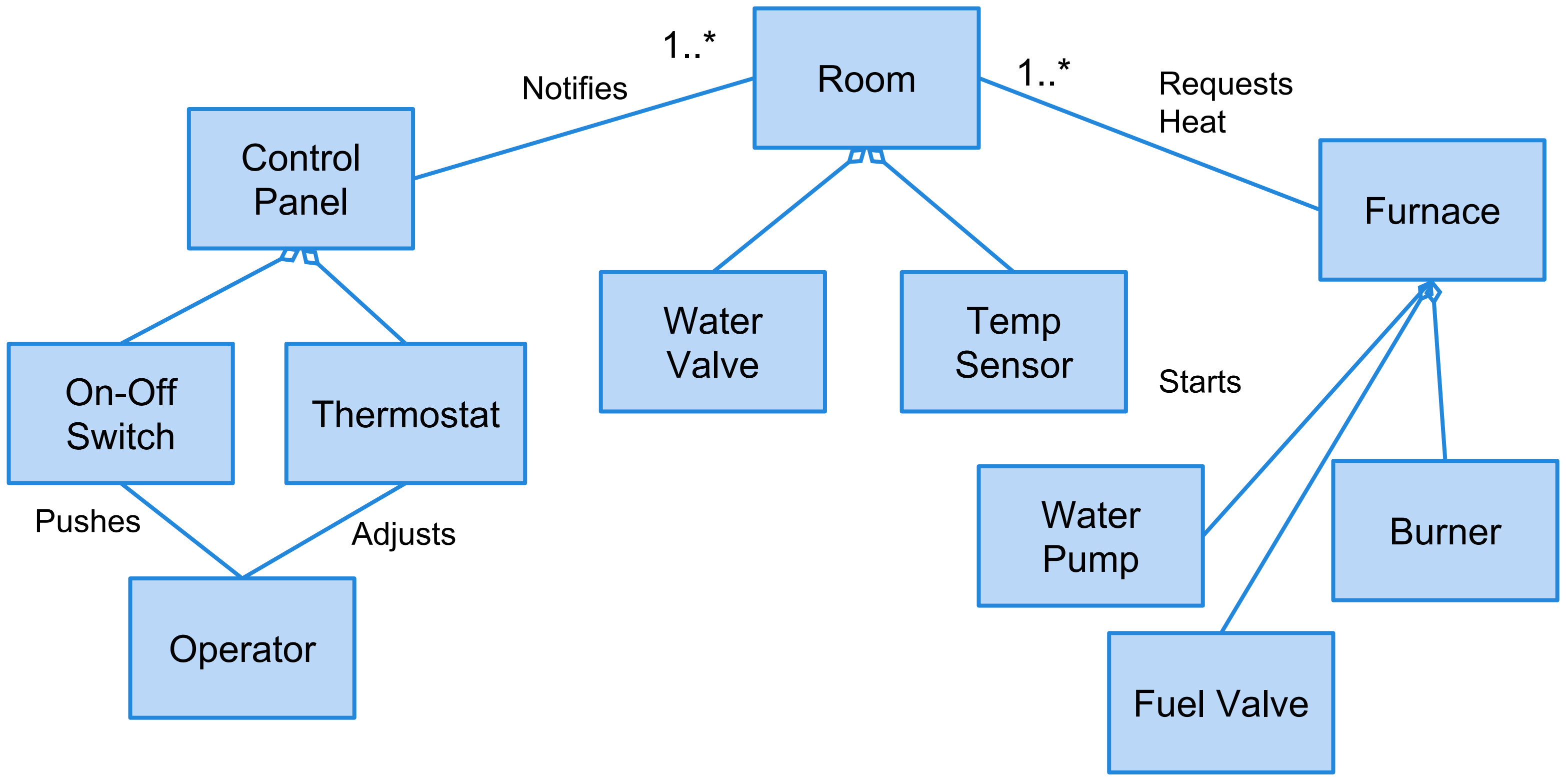
Refinement 1 - Better?



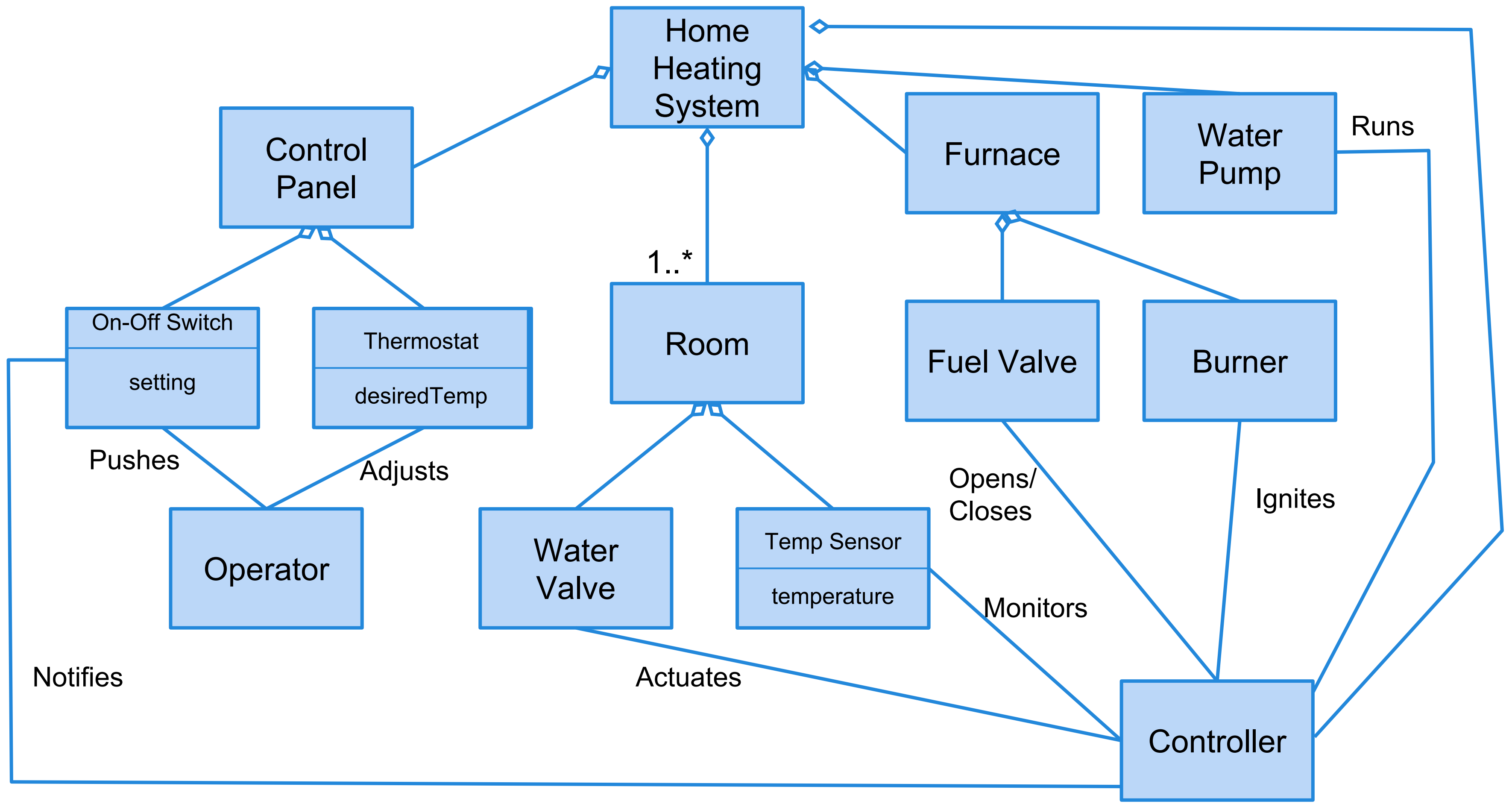
Class Diagram - Round 2



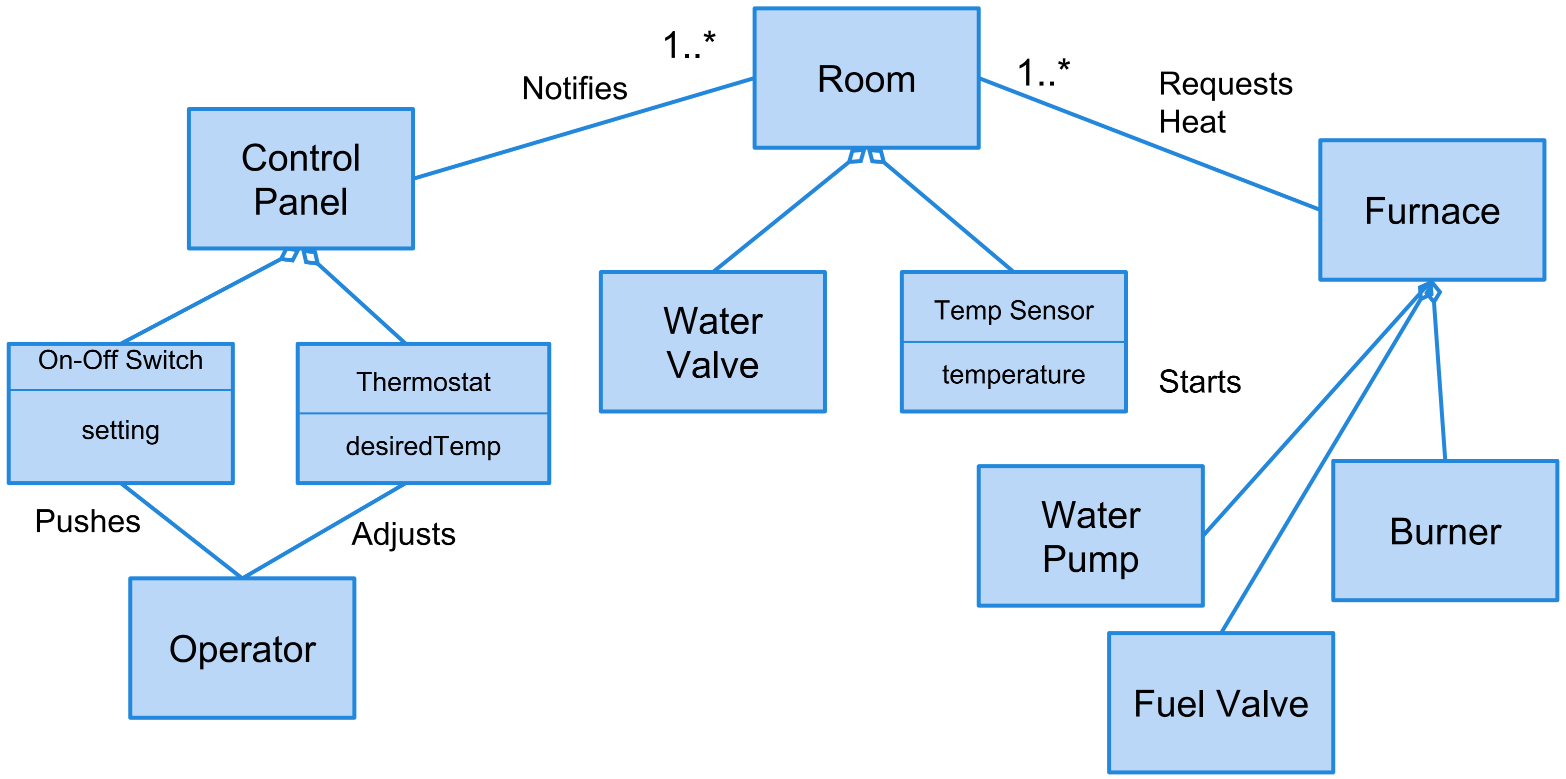
Class Diagram - Alternate



What about Attributes?



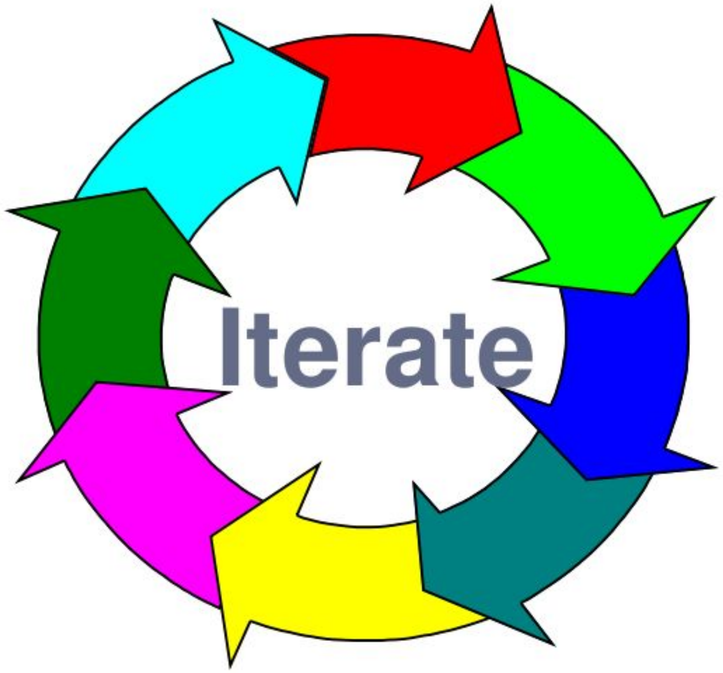
Attributes - Alternate



Iterate the Model

Keep on iterating until you, your customers, and your engineers are happy with the design.

Any questions on class diagrams?



We Have Learned

- How to approach an OO modeling effort
  - Identify objects (nouns)
  - Identify operations and associations (verbs)
  - Identify attributes.
  - Refine, refine, refine!
- The model will need a lot of iteration.
  - And often requires a dynamic view of the system as well (we'll get to that soon).

Next Time

- Design Patterns
  - Design advice for common scenarios.
- Reading
  - Sommerville, chapter 7
- Work on your design for BILL.
- Questions?