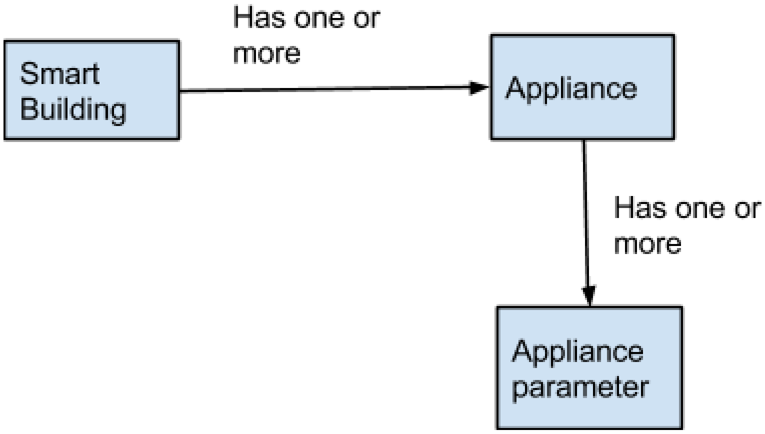
CSL202 | Assignment#2 | 14th Feb 2014 | Due on 21st Feb 2014

Important Instructions

1. Plagiarism and sharing code will lead to F grade (and possibly disciplinary action for repeat cases). You are responsible for ensuring that you code is not copied by others.
2. You may use tools such lex/yacc or ANTLR or PyParse etc. to generate grammar parser. Alternatively, you can also hand code such a grammar parser yourself.
3. Please document the code properly; there is partial credit for clean and well written code.
4. Your score for an assignment solution will be assessed by looking at:
    1. Implementation approach and its correctness.
    2. Readability of the code and associated manual/readme etc.
    3. Correct functioning of the code. That is, it should produce correct results for various input scenarios.
    4. Quality of your design and code. For example, if changing one input value requires you to recompile your program then it is a bad design.

Description

In this assignment, we first need to build a grammar to allow expressing the configuration of a smart building. And also create a program which reads such a configuration and creates necessary objects to simulate the scenario. Configuration of a smart building includes specifying what different types of appliances are installed in the building. Clearly, this will require the grammar to allow expressing the configuration of individual appliances as well. Following diagram illustrates the major entities and interrelationships among them.



Smart Building

A smart building may have one or more appliances such as AC, lighting, fan, ambient temperature sensor, ambient light sensor etc.

Appliances

An appliance must support two operations:

1. `sense(param p)`: It senses and returns the specified parameter value of the appliance. For example, temperature from ambient temperature sensor, or current cooling setting of an AC.
2. `control(param p, value v)`: It controls the specified parameter value of the appliance. For example, current cooling setting of an AC.

Appliance Parameter

An appliance may have only a specific set of meaningful parameters which are available to external entities for sensing and control. Each parameter will have some sensible range and type. For instance, a valid range of values for a tube light state could be ON an OFF. It could be represented by either a string (“ON” or “OFF”) or as integers (0 for off and 1 for on state) etc.

Grammar for Building Configuration

In order to simulate a practical scenario we need to be able to specify the configuration in a easy to read and write language. For example, one should be able to write a statements script like:

Line #1. AC IS AN APPLIANCE WITH PARAMETERS [Cooling Level:(0-40), AH Level:(0-100), State:(“ON”,”OFF”)]
Line #2. Tube Light IS AN APPLIANCE WITH PARAMETERS [State:(“ON”,”OFF”)]
Line #3. Light Sensor IS AN APPLIANCE WITH PARAMETERS [Ambient Lux:(0-100000)]
Line #4. Fan IS AN APPLIANCE WITH PARAMETERS [RPM:(0-5000), Motor Temperature:(0-200)]
Line #5. L5 = Class Room IS A BUILDING HAVING [AC:3,Tube Light:10]
Line #6. JUP = Jupiter Hostel IS A BUILDING HAVING [Tube Light: 150, Fan:100, Light Sensor, 2]
Line #7. // Operate on the building and appliances
Line #8. CONROL RPM OF JUP FAN[1] AS 2000
Line #9. myAcState = SENSE State OF L5 AC[2]

With the statements like above one should be able configure a scenario involving a building having some kind of appliances. When a script such as the above is fed as input the “interpreter/compiler” that you create, it should instantiate necessary objects for building and appliances.