

SE 3354 Homework #4, Design

A system periodically gets data from two different sensors, a radar and an infrared sensor.

There is general sensor information that is common to all types of sensor, plus unique attributes of each type of sensor.

The sensor data is fused to provide an integrated view of the external environment, which is shown on a display.

The user can adjust various controls to affect the information displayed.

- a) What objects can you identify for this system? (20 points)**
- b) What attributes (if any) can you identify to characterize each object? (15 points)**
- c) What methods (if any) can you identify for each object? (15 points)**
- d) Draw a UML class diagram to capture a static view of the system. (25 points)**
- e) Draw a UML sequence diagram to capture a dynamic view of this system. (25 points)**

Grading Rubric

Should be one class diagram and (at least) one sequence diagram.

Classes in class diagram should match objects in sequence diagram.

Classes need to have a reasonable set of attributes and methods.

Should have associations between classes in class diagram.

Associations in class diagrams should match messages in sequence diagrams.

4a) Objects

A system periodically gets data from two different sensors, a radar and an infrared sensor. The sensor data is fused to provide an integrated view of the external environment and displays the view on a display. The user can adjust various controls to affect the information displayed.

Nouns

- **user == actor not object**
- **system (fuser?)**
- **sensor**
 - radar
 - infrared
- **data**
 - attributes of sensors?
- **view, environment, display, information**
 - operationally, synonyms?
- **controls (noun)**

system “fuses” the data

radar object + infrared object

- **could do as base class sensor with two derived subclasses, radar and infrared**
 - **generalization / specialization**
 - **superclass / subclass**
 - **base class / derived class**

display

- **a view of the environment**
- **contains “information” (fused from sensors)**

controls → controller

Objects

Two sensors

- radar
- infrared

Display

“Fuser”

- conceptual object

Controls

4b) Attributes

No specifics in this very high-level problem statement

“System” fuses data from the sensors

- **fused data is what? combined? separate but equal? whatever is needed by display?**

Sensors have sensor data

- **different for radar and infrared, but don't know how different**

Display has view of the environment

Controller – fusion settings? display settings?

Attributes

Two sensors

- **general data for sensors**
- **radar-specific data**
- **infrared-specific data**

Display

- **display information**
- **display settings**

“Fuser”

- **data used in fusing data**
- **fusion settings**

Controls

- **control settings**

4c) Methods

Have to make assumptions about how the system works

- **does the “main” poll the sensors for data?**
- **do the sensors interrupt with data?**
- **sensors and main coordinate through a shared memory?**

Assuming a polling design – fairly common in this kind of system...

Assuming that the Fuser object is the driving the system

- **could also be controller object**

Sensors will have a method that can be used to access the data they are getting from the hardware

- **getRadarData**
- **getIRData**
- **includes generic sensor data from Sensor superclass by implication**

Sensors will probably have private methods for getting the data from the hardware

- **senseRadar**
- **senseIR**

Fuser – the “main program” – will have call the sensor methods to get data then fuse it

Fuser will call the controller to see if settings have been changed

Fuser will call the display

- **Display will need a method to change the view on the screen: displayData**

Display will need a method to change what information is displayed on the screen

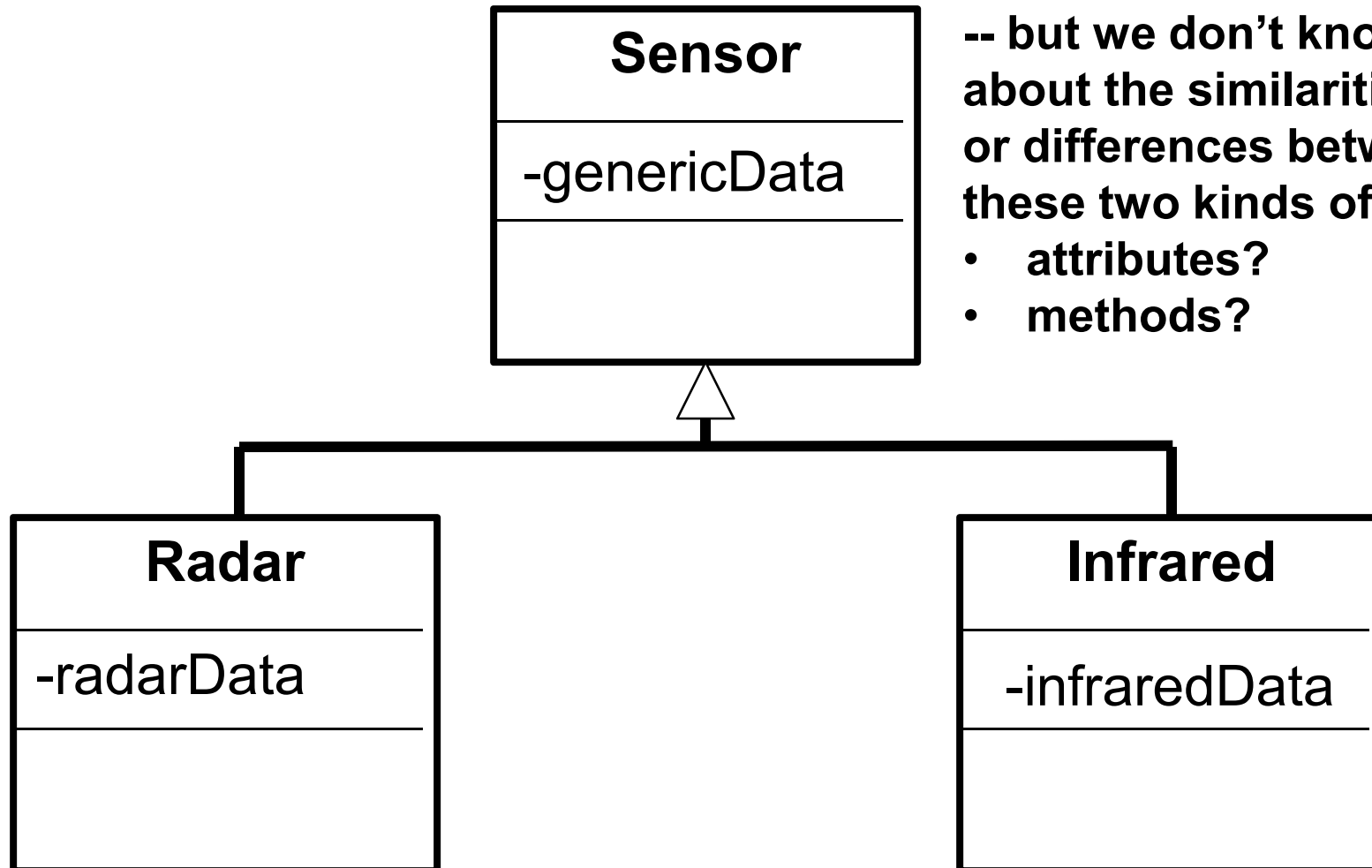
- **changeSettings**

Assuming that the main program is Fuser, then it will poll the controller routine to see if the user is changing the settings

Controller will need a method to see if the control settings for fusing or displaying have been changed

- **for high cohesion, should have two methods**
- **one for changing display settings: tweakDisplay**
- **one for changing fusing settings: tweakFusing**

4d) Classes

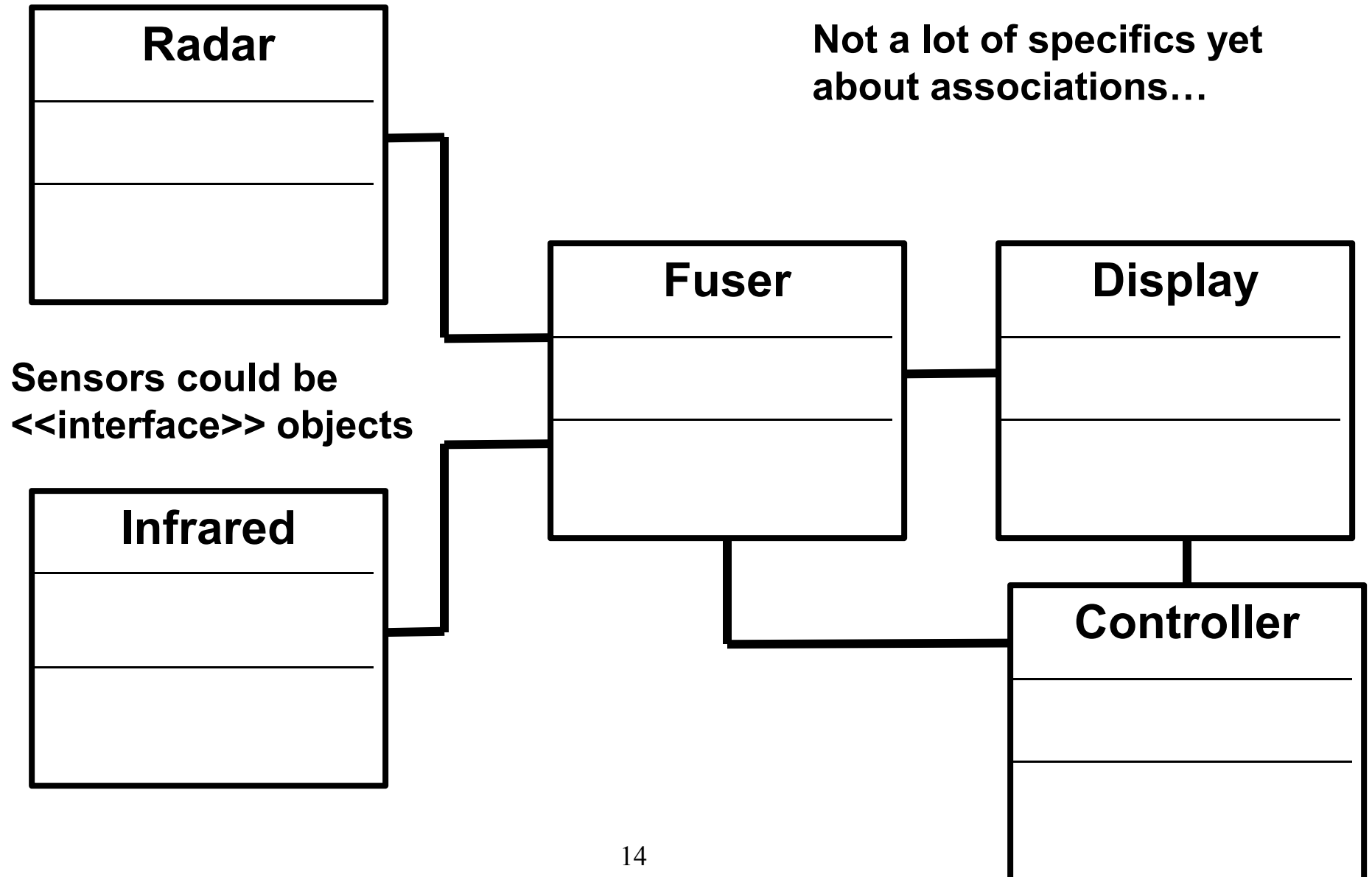


Implied doing as superclass
sensor with subclasses
for radar and infrared
-- but we don't know much
about the similarities
or differences between
these two kinds of sensor

- attributes?
- methods?

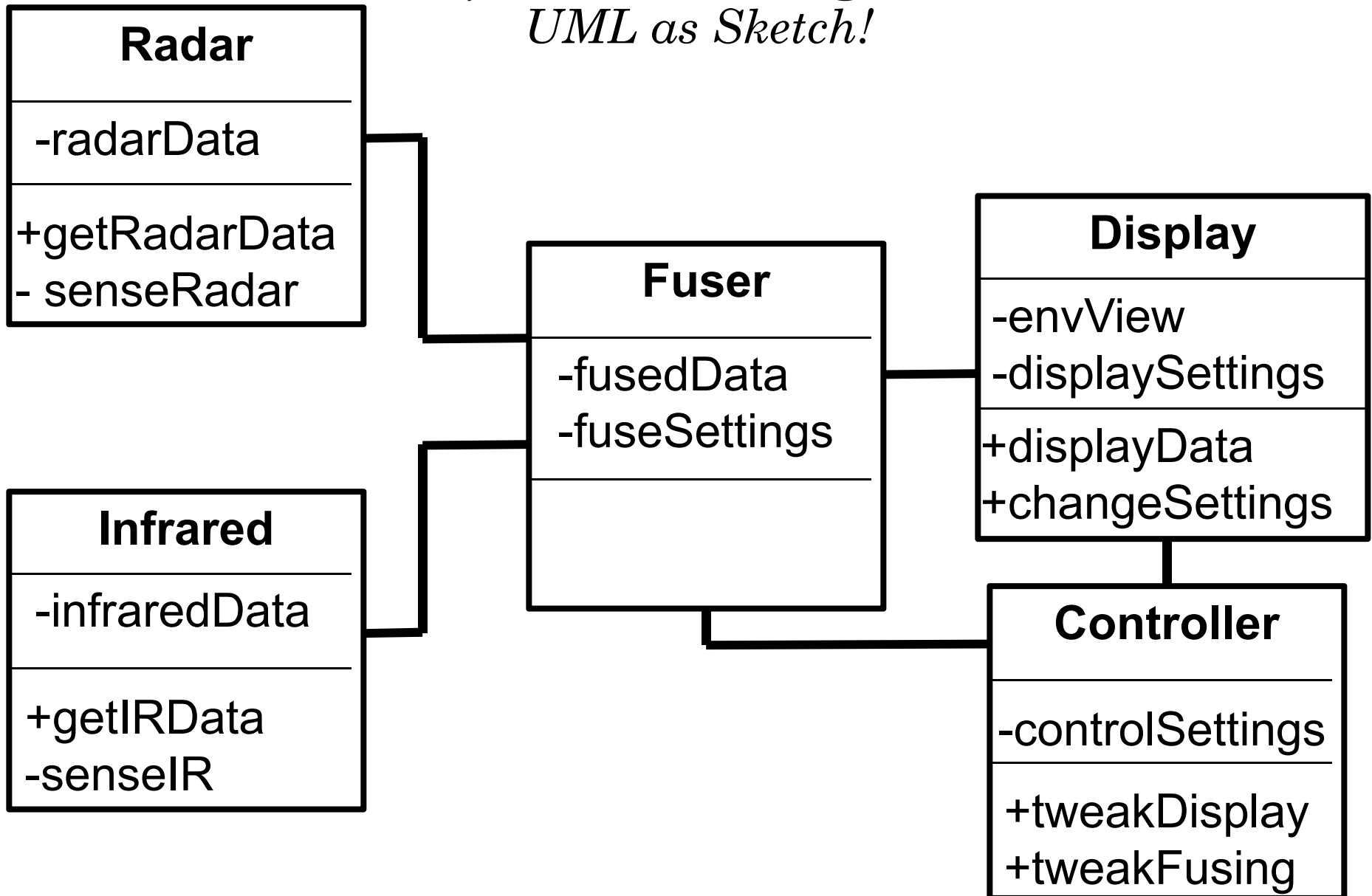
**Radar and infrared as
concrete objects**

**Not a lot of specifics yet
about associations...**

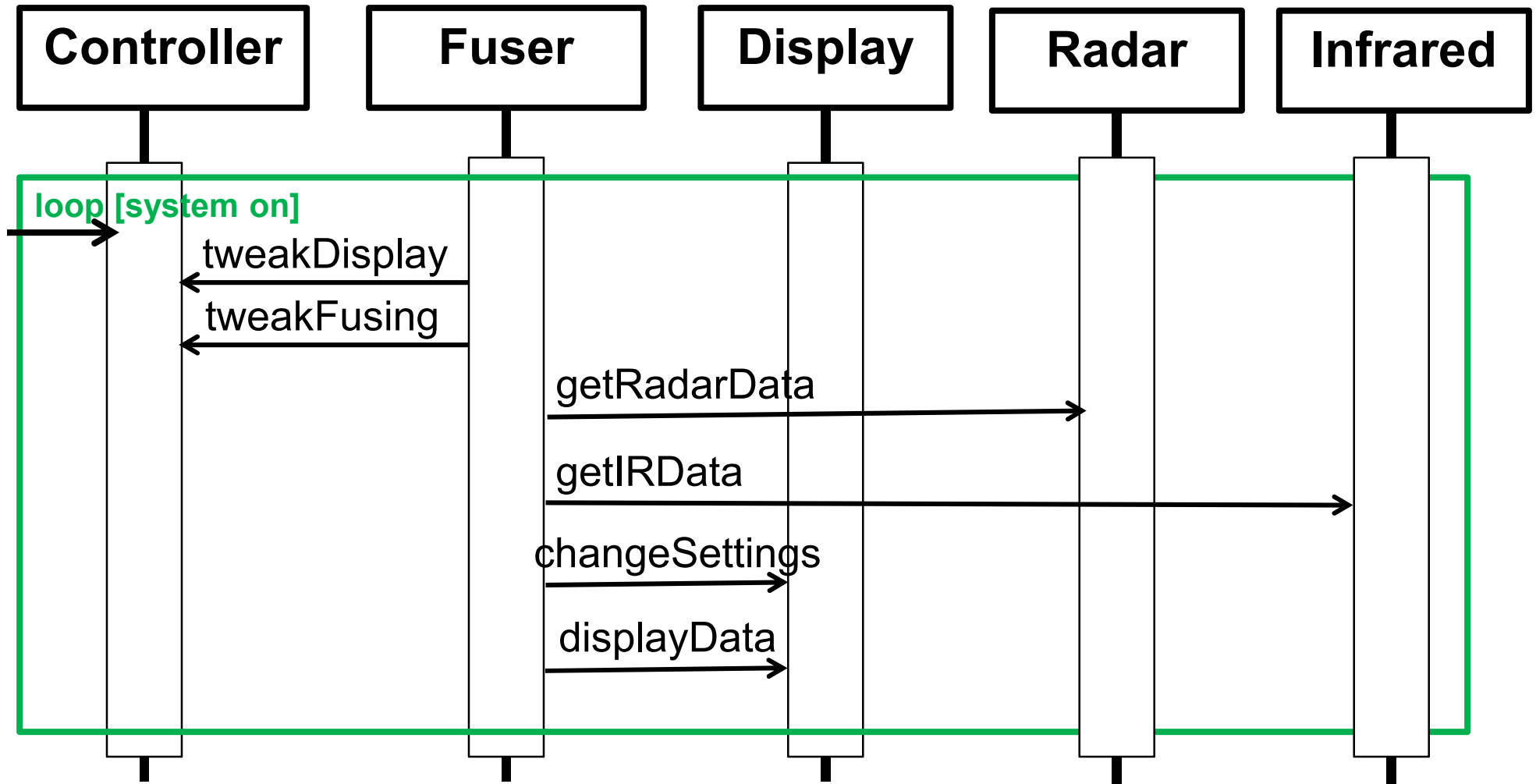


4d) Class Diagram

UML as Sketch!



4e) Sequence Diagram



Some actions are occurring “in the hardware” associated with the sensors and the controller

- **controlSettings**
- **senseRadar**
- **senseIR**

Infinite loop for this embedded system

MVC is implied pattern from class diagram

Detailed Grading Rubric

4.1 Identifying five objects + (optional) superclass, 20 points, 4 points each

- might have identified more (within reason) though it's hard to see exactly what they would be
- could have sensors as <<interface>> objects

4.2 Attributes, 15 points

Sensor data (radar + infrared), 2 points each

Fused data, fused settings, 2 points each

Display data (envView), display settings, 2 points each

Controller settings, 2 points

- any other objects should have appropriate attributes (may be none depending in what the object is), -1 point if inappropriate

4.3 Methods, 15 points

- depends on how you designed the system given whatever assumptions you made
- examples discussed are not exhaustive

4.4 Class diagram, 25 points

- **should have reasonable attributes and methods identified**
- **should match objects from 4.1, 10 points**
 - only count off once in 4.1 for any “bad” object identifications
- **many assumptions must be made about how the system works – anything plausible is allowed**
- **classes should have associations with other classes, 5 points**
 - need not have name, directionality, cardinality since I’m emphasizing UML as sketch

4.5 Sequence diagram, 25 points

- **should show reasonable message passing between objects**
- **should have reasonable match with objects in class diagram**
 - any significant mismatch is worth 25 points across these two parts
- **messages may or may not be named since I'm emphasizing UML as sketch**
 - note that in a more complex design, naming messages and associations would be highly desirable
- **do not worry overly much about synchronous vs asynchronous and similar syntactical issues**
 - UML as sketch...