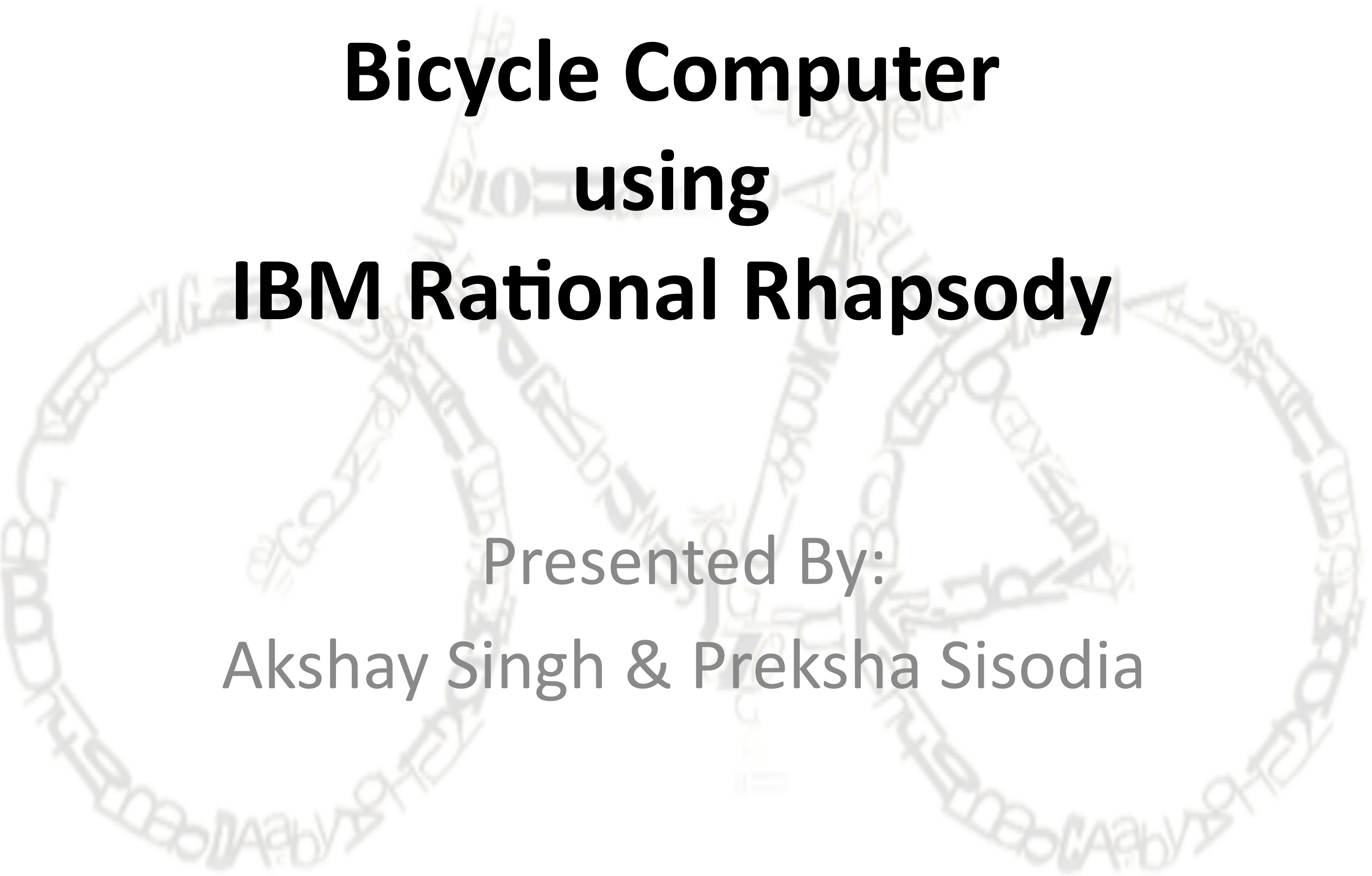
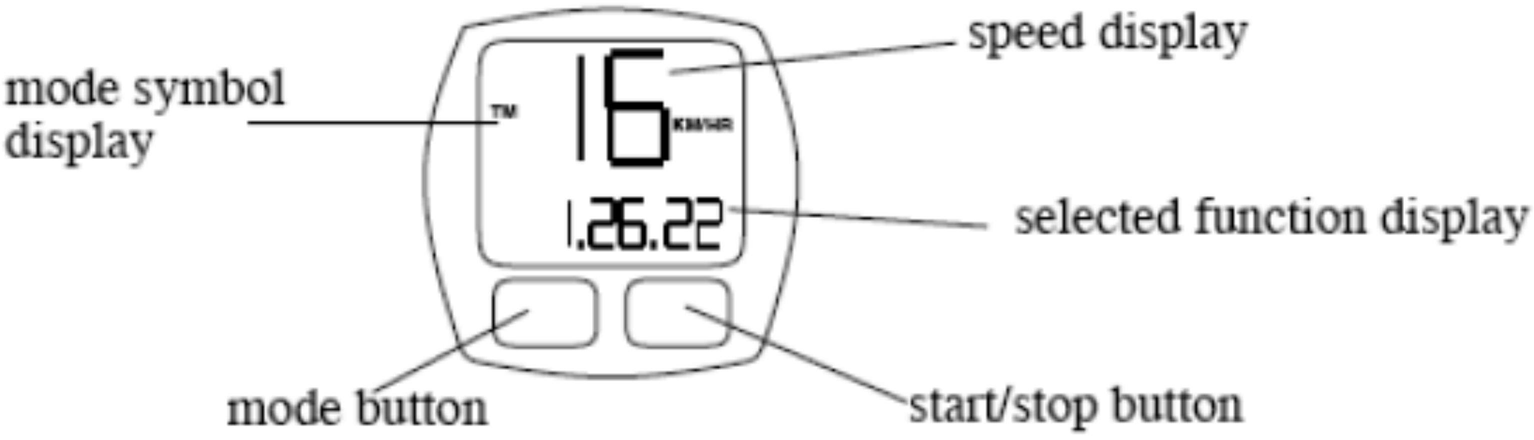
Bicycle Computer
using
IBM Rational Rhapsody



Presented By:

Akshay Singh & Preksha Sisodia

Problem Statement



To design an extendible model

MBSE Technology Used

IBM Rational Rhapsody

- IBM Rational Rhapsody Developer V 7.6.1.0

- Supports:
  - ✓ C/C++ ✓C# ✓Java ✓ Ada

- Provides various techniques (Profiles, stereotypes, etc) to extend basic functionalities

Features of Rational Rhapsody

- Use of profiles
  - Can be used to design DSMLs
  - Provides good starting point

- Full support for Round-Trip Engineering

- Animation for model validation
  - Animated behavioral diagrams

- Manage compatibility with previous product versions

Features of Rational Rhapsody (2)

- Integrates well with other Rational Jazz Platform tools
  - Rational Team Concert
  - Rational Synergy

- Collaboration
  - Central web interface

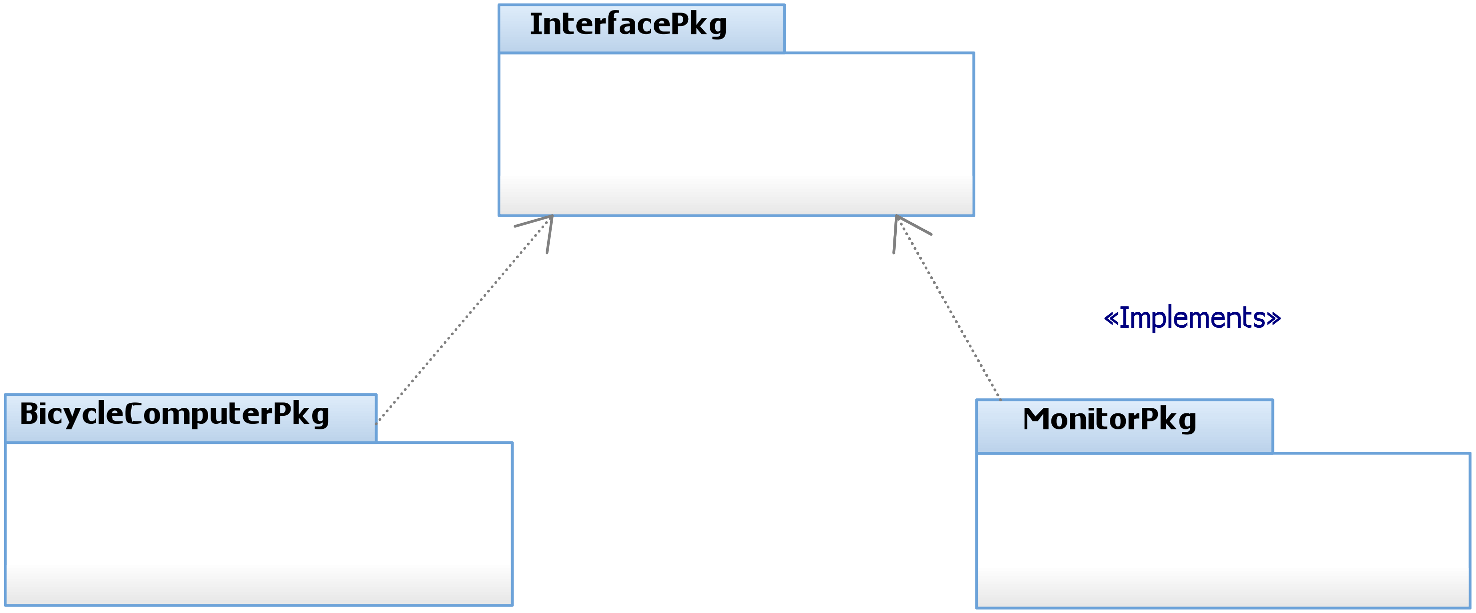
- Improvements in 7.6.1
  - Support AUTOSAR
  - Integration of MathWorks Simulink on Jazz

Our Methodology

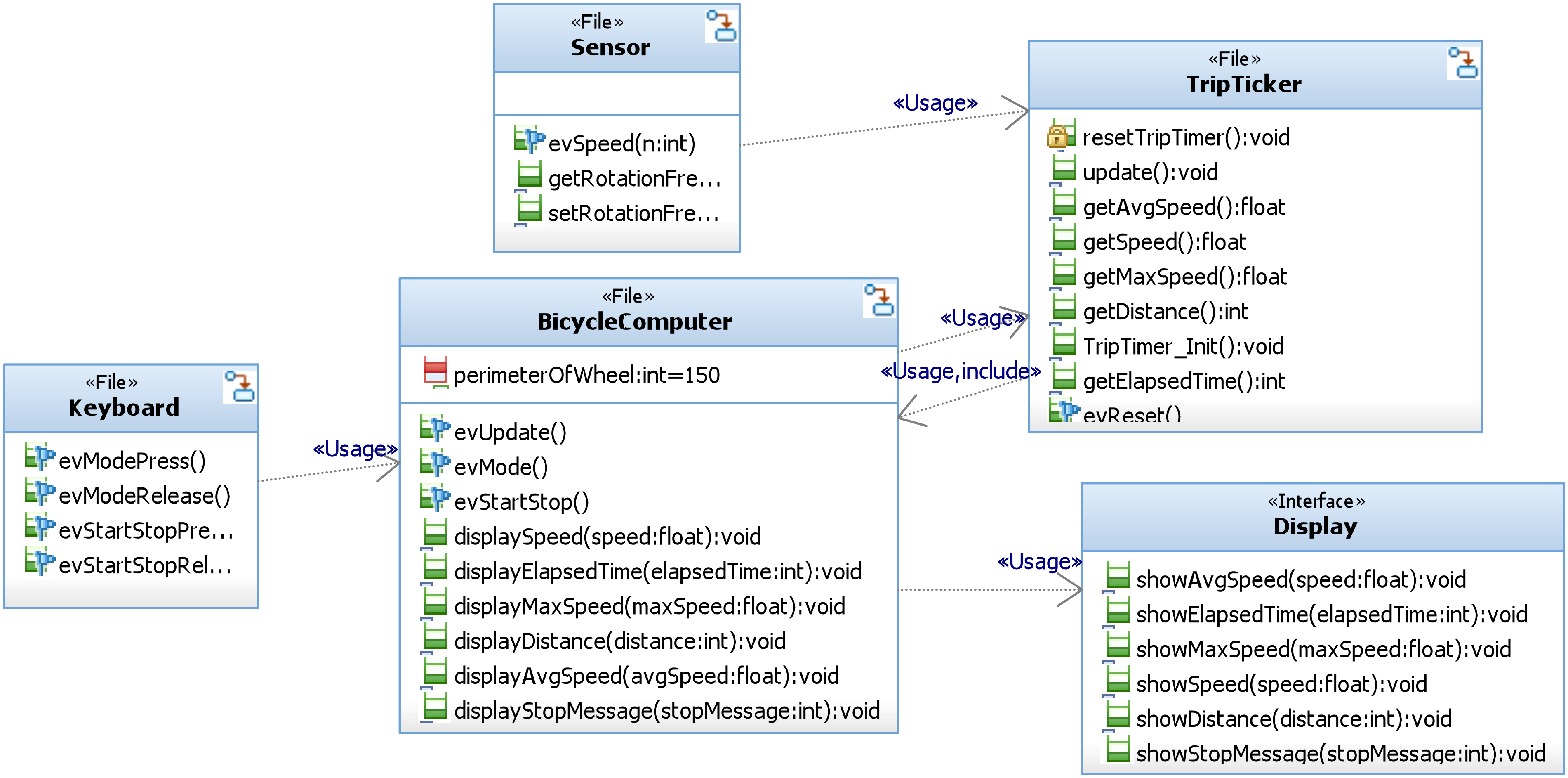
- started with VS-2010, not supported yet
- Rhapsody perspective in Eclipse
- Implemented using internal Rhapsody internal GUI

Our Methodology (2)

Functionality is divided in to three main packages



Core Functionality



«File»
Sensor

evSpeed(n:int)
getRotationFre…
setRotationFre…

«Usage»

«File»
TripTicker

resetTripTimer():void
update():void
getAvgSpeed():float
getSpeed():float
getMaxSpeed():float
getDistance():int
TripTimer_Init():void
getElapsedTime():int
evReset()

«Usage»

«Usage,include»

«File»
Keyboard

evModePress()
evModeRelease()
evStartStopPre…
evStartStopRel…

«Usage»

«File»
BicycleComputer

perimeterOfWheel:int=150

evUpdate()
evMode()
evStartStop()
displaySpeed(speed:float):void
displayElapsedTime(elapsedTime:int):void
displayMaxSpeed(maxSpeed:float):void
displayDistance(distance:int):void
displayAvgSpeed(avgSpeed:float):void
displayStopMessage(stopMessage:int):void

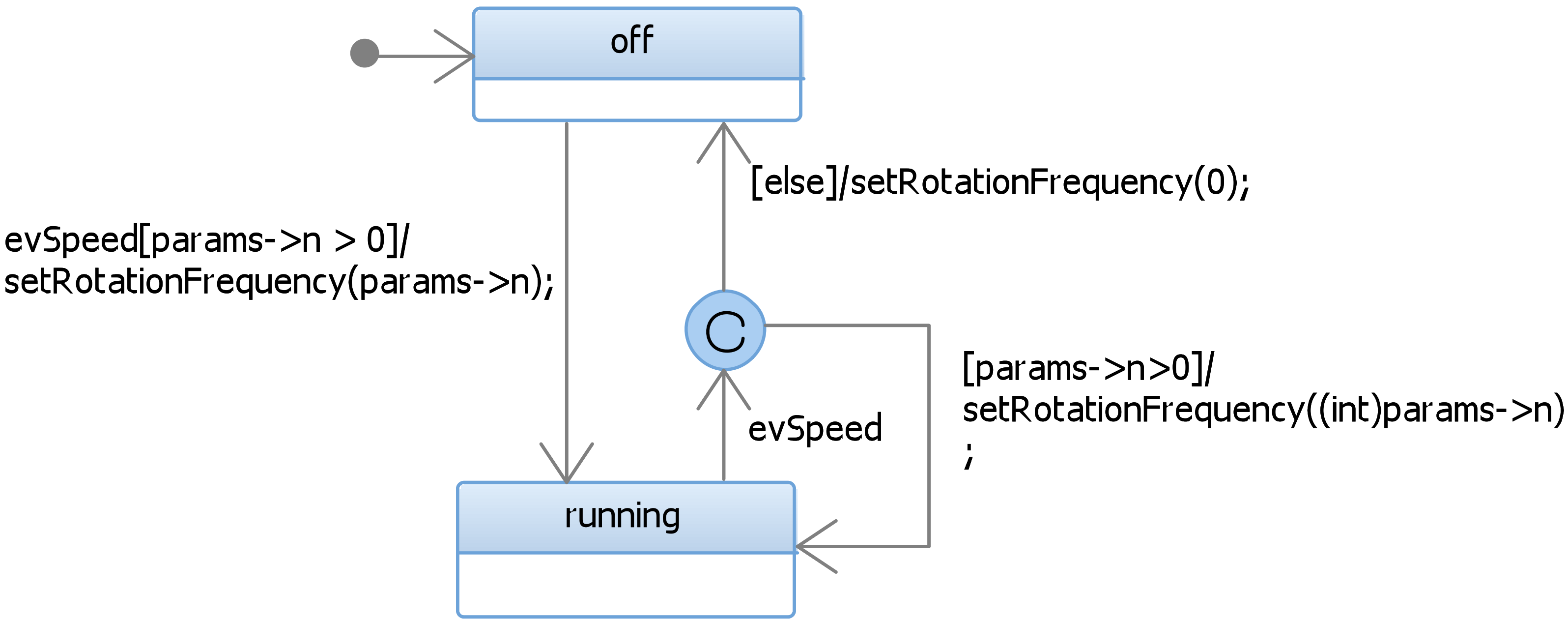
«Usage»

«Interface»
Display

showAvgSpeed(speed:float):void
showElapsedTime(elapsedTime:int):void
showMaxSpeed(maxSpeed:float):void
showSpeed(speed:float):void
showDistance(distance:int):void
showStopMessage(stopMessage:int):void

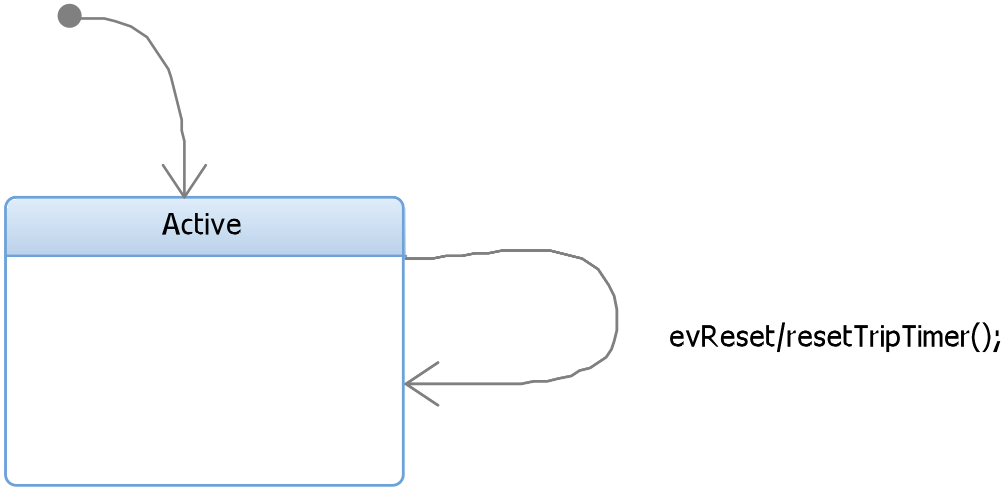
Main Object Model Diagram

Putting the logic in Model


off
[else]/setRotationFrequency(0);
evSpeed[params->n > 0]/
setRotationFrequency(params->n);
C
[params->n>0]/
setRotationFrequency((int)params->n)
;
evSpeed
running

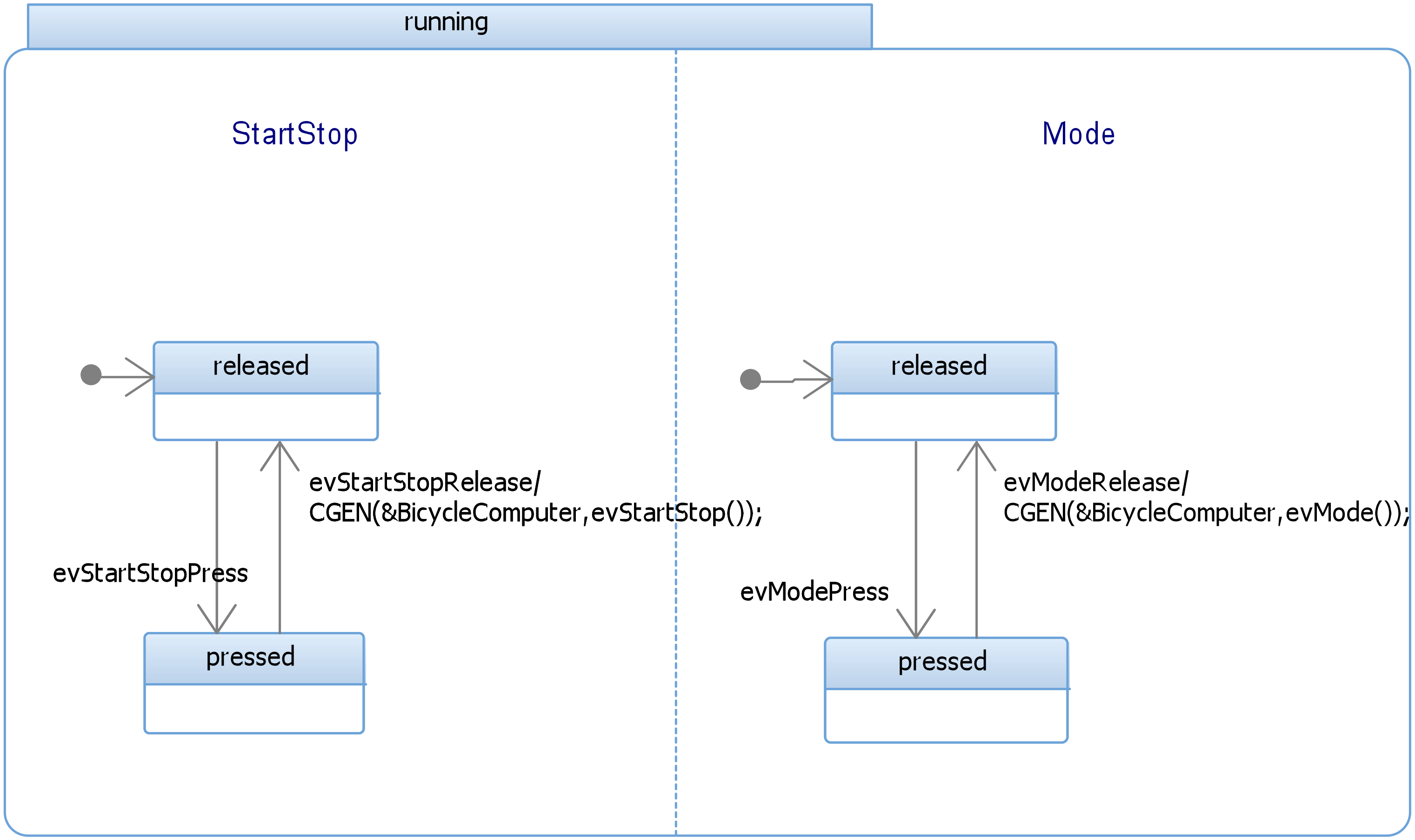
Statechart for Sensor Class

Putting the logic in Model (2)



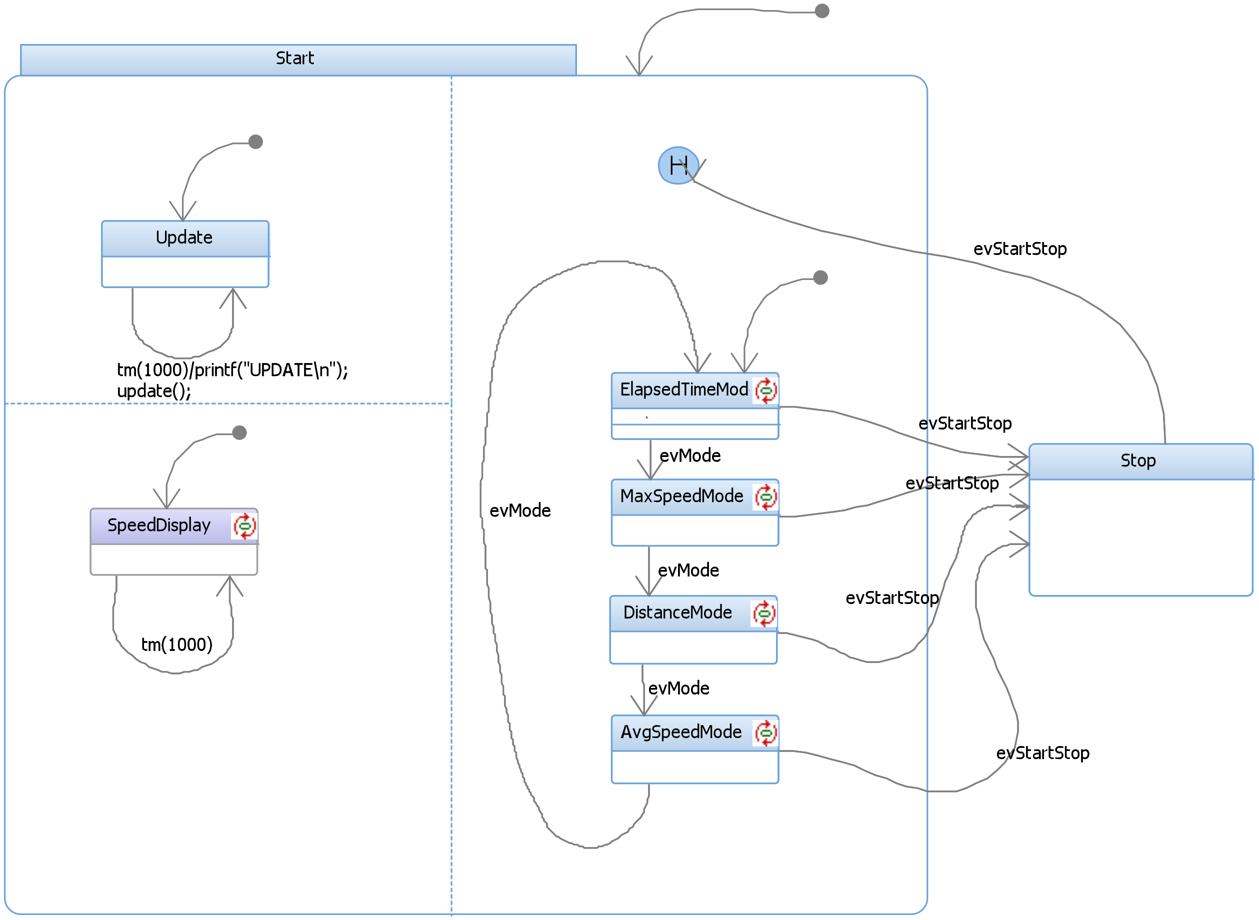
Statechart for TripTicker Class

Putting the logic in Model (3)



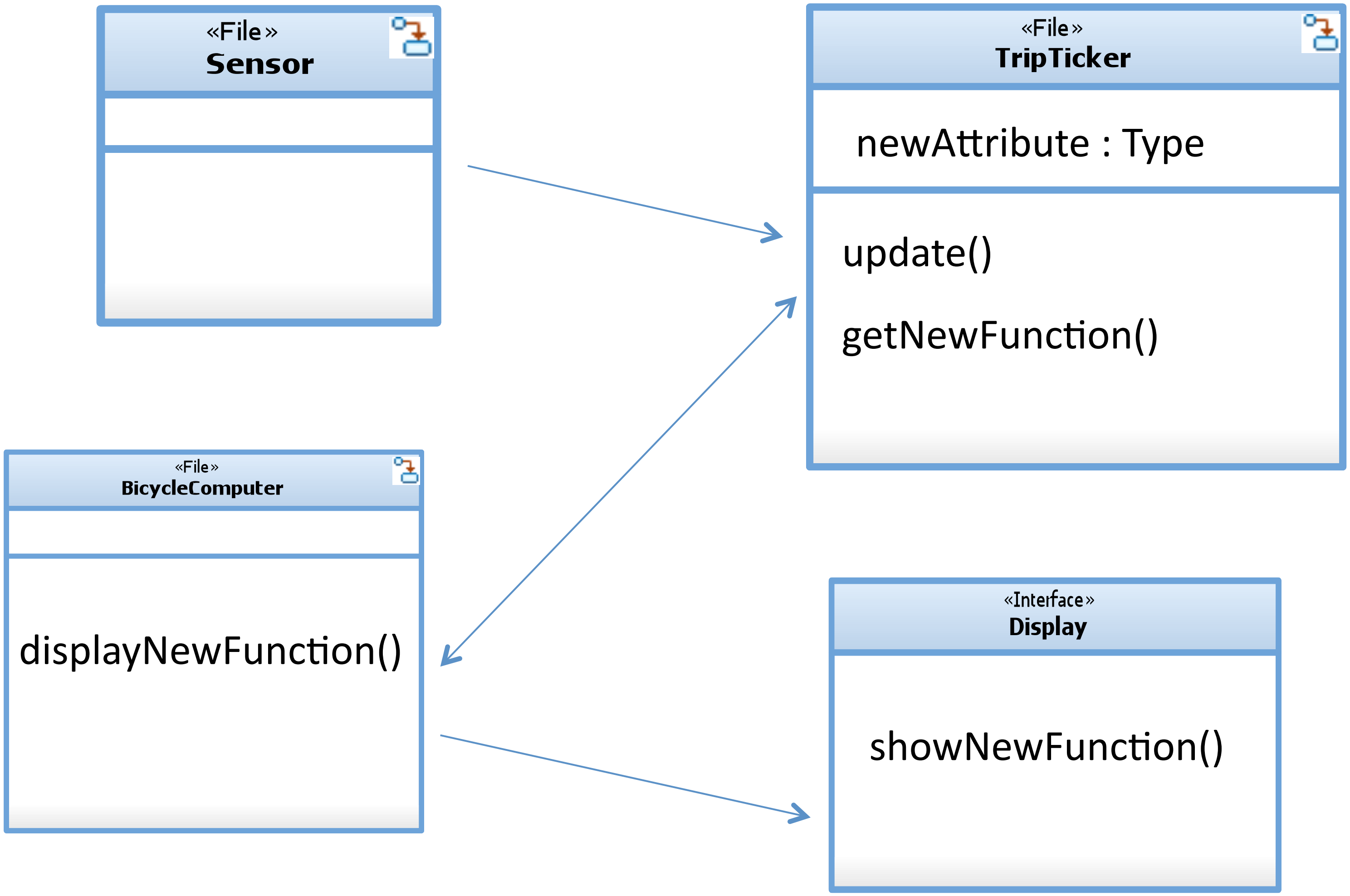
Statechart for Keyboard Class

Putting the logic in Model (4)



Statechart for BicycleComputer Class

How to add new functionality ?



«File»
Sensor

«File»
TripTicker

newAttribute : Type

update()

getNewFunction()

«File»
BicycleComputer

displayNewFunction()

«Interface»
Display

showNewFunction()

DEMO

(adding a new mode function for displaying Calories)

Insights

- Fairly simple tool for UML modelling
- Clean integration of model and code
  - show code locally
  - show code in IDE
- Ability to merge state-charts and activity diagrams

Insights(2)

- Great error detection and debugging features
  - Detailed and linked error messages
  - Animated trace for behavioral diagrams

- Round-Trip Engineering is not very handy with multiple Components

Lesson Learned

- Start with online available examples, instead of following formal documentation
- First look for available profiles before starting development
- Configuration gets associated with attributes & operations, and also with closing/opening a project
  - **Solution**: we may need to delete and again create those attributes/operations

Lessons Learned (2)

- For external GUI integration (e.g. VS, eclipse)
  - Import is better than export

- Very poor performance with Linux
  - No rhapsody perspective for eclipse

- Expect your license after project deadline!!

Thanks!