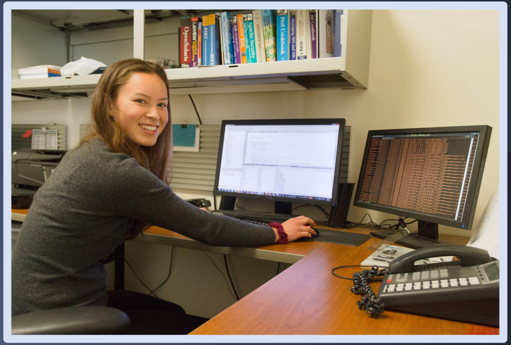
Calculating Specialty Materials Usage Accurately: an Improved Algorithm and a Regression Series Mapping Voltage to Flow Rates

Rachel Barcklay
Berkeley High School, Summer 2015

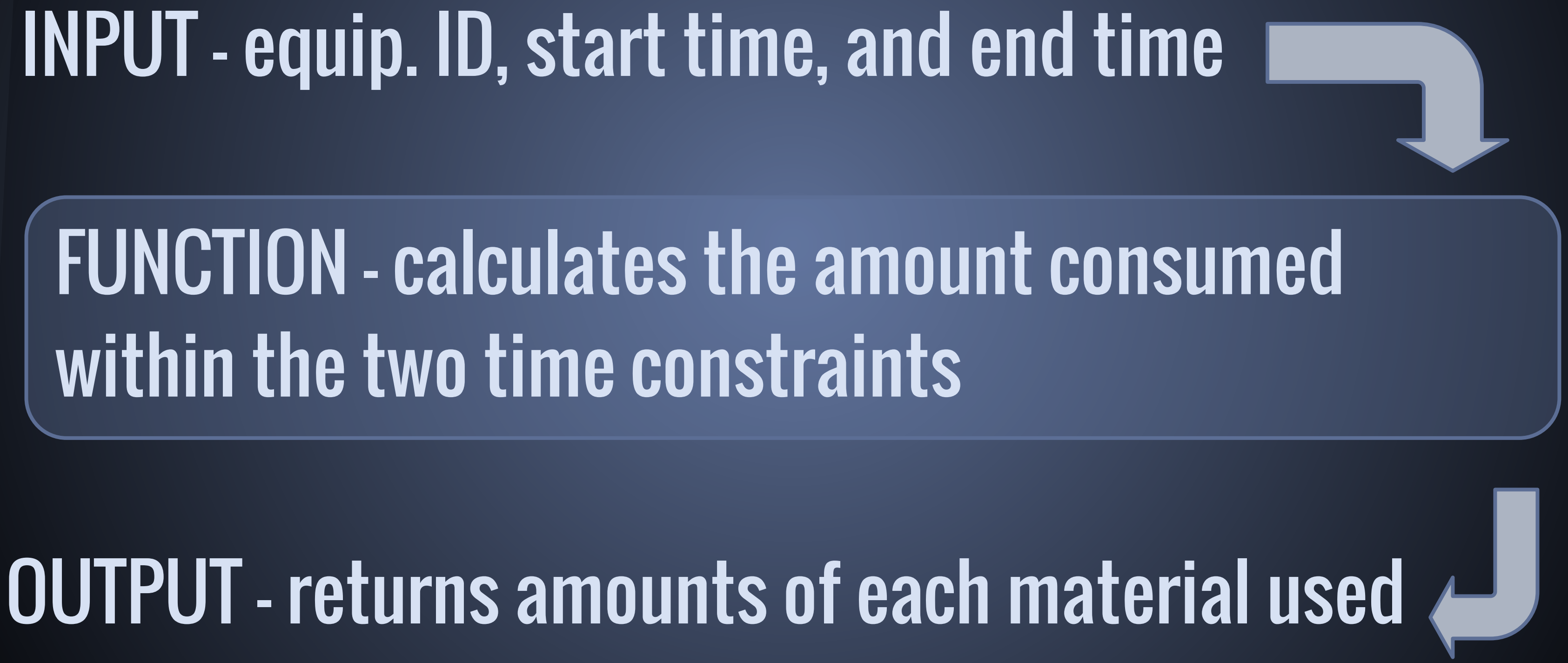
Marvell Nanolab

Presentation Agenda

- programming project
    - overview
    - implementation
- lab work
    - objectives
    - results



RUMS Usage Method Overview



INPUT - equip. ID, start time, and end time

FUNCTION - calculates the amount consumed within the two time constraints

OUTPUT - returns amounts of each material used

RUMS Usage Method Implementation

- uses **parameterized queries** (vs. string concatenation)

Parameterized Query

- checked input
- prevents most SQL injection attacks

String Concatenation

- unchecked input
- less secure than prepared statements

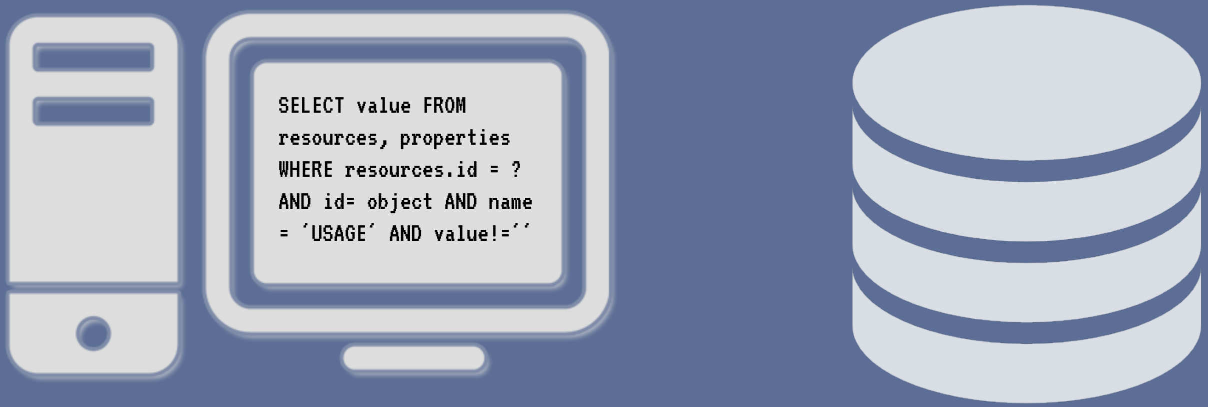
SELECT id FROM equipment WHERE name = “?, DROP TABLE equipment”

What next?

- gets back information about all sensors associated with the equipment
  - sensor names & ID's
  - which gases the sensors monitor
  - **sensor types**

Which sensors are connected to the AMATEPI?

Germane_flow_Epi & Dichlorosilane

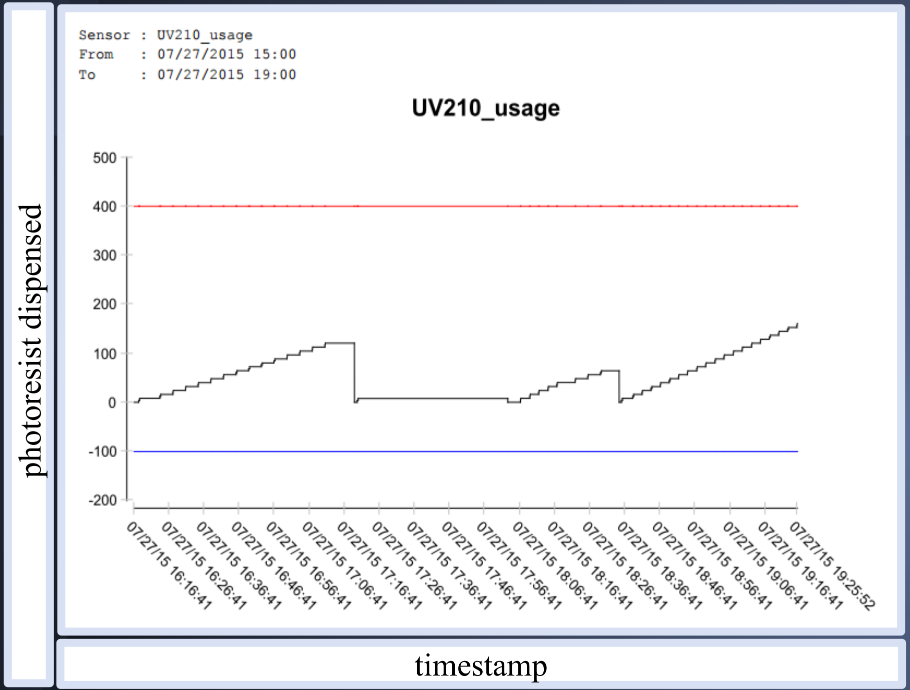


```
SELECT value FROM
resources, properties
WHERE resources.id = ?
AND id= object AND name
= ´USAGE´ AND value!=´´
```

Mercury

RUMS Usage Method Implementation cont...

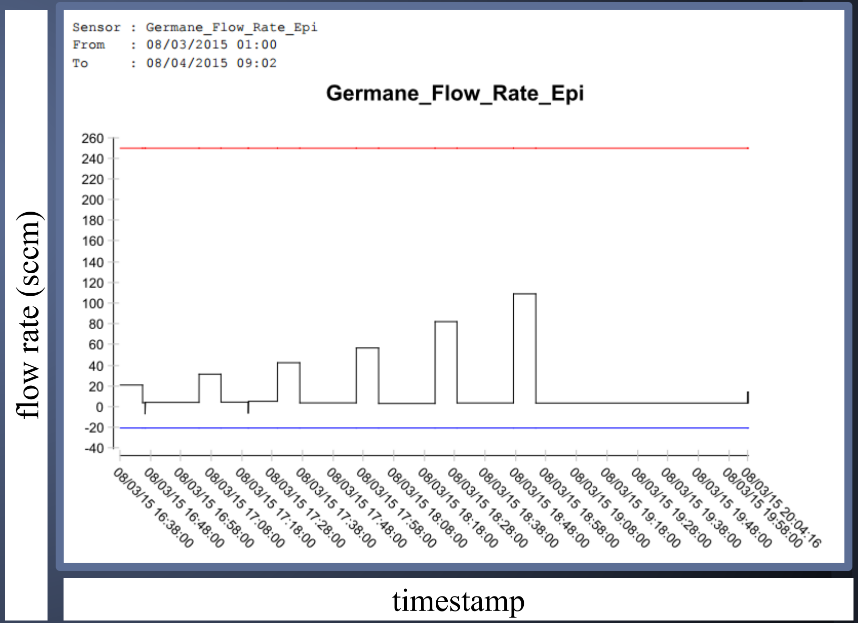
- queries RUMS (parameters are sensor ID, start time, end time)
- statement returns data points for the sensor between start and end times



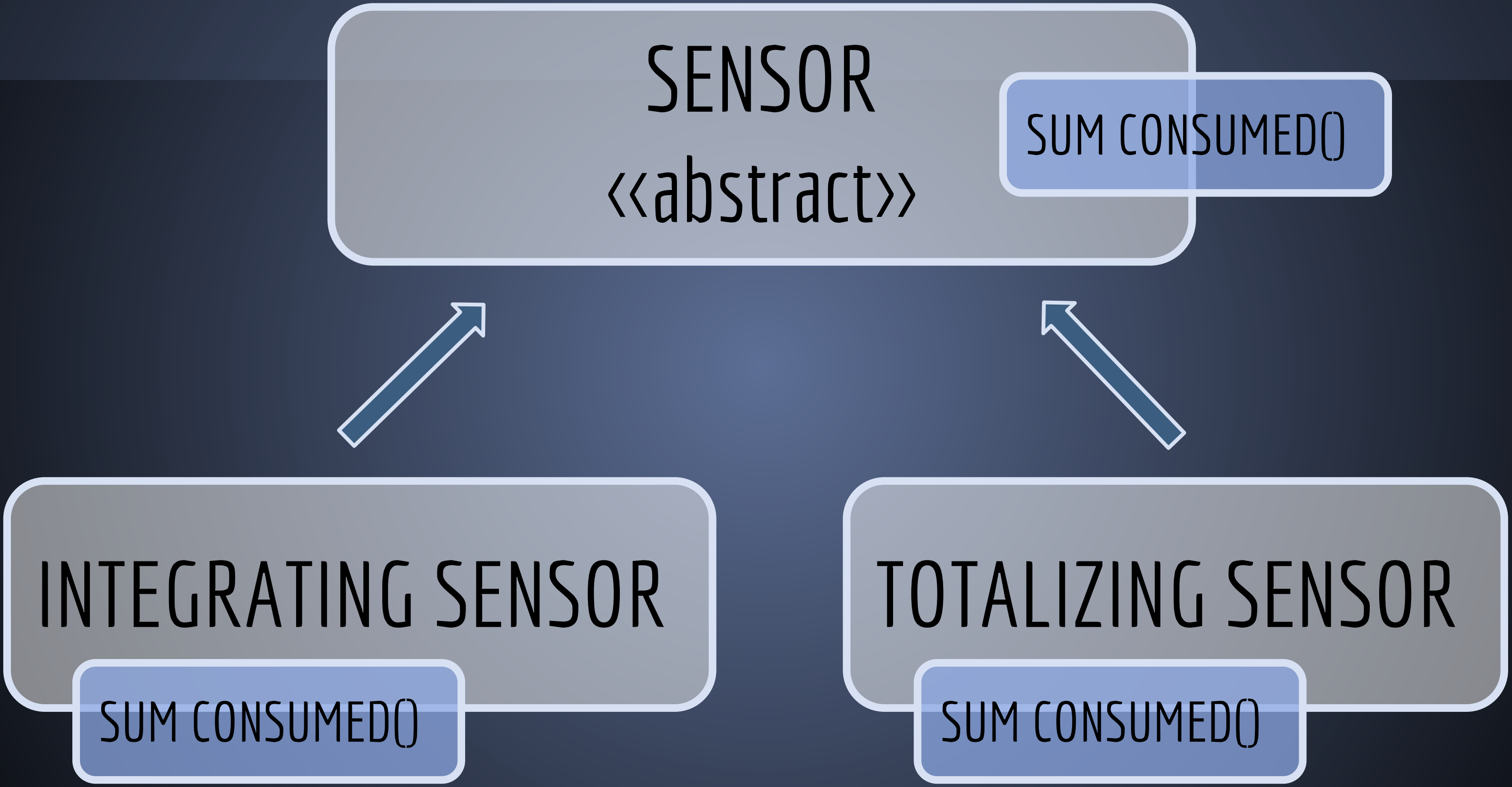
Sensor : UV210_usage
From   : 07/27/2015 15:00
To   : 07/27/2015 19:00

Totalizing sensor

Integrating sensor



Sensor : Germane_Flow_Rate_Epi
From   : 08/03/2015 01:00
To   : 08/04/2015 09:02



SENSOR
«abstract»

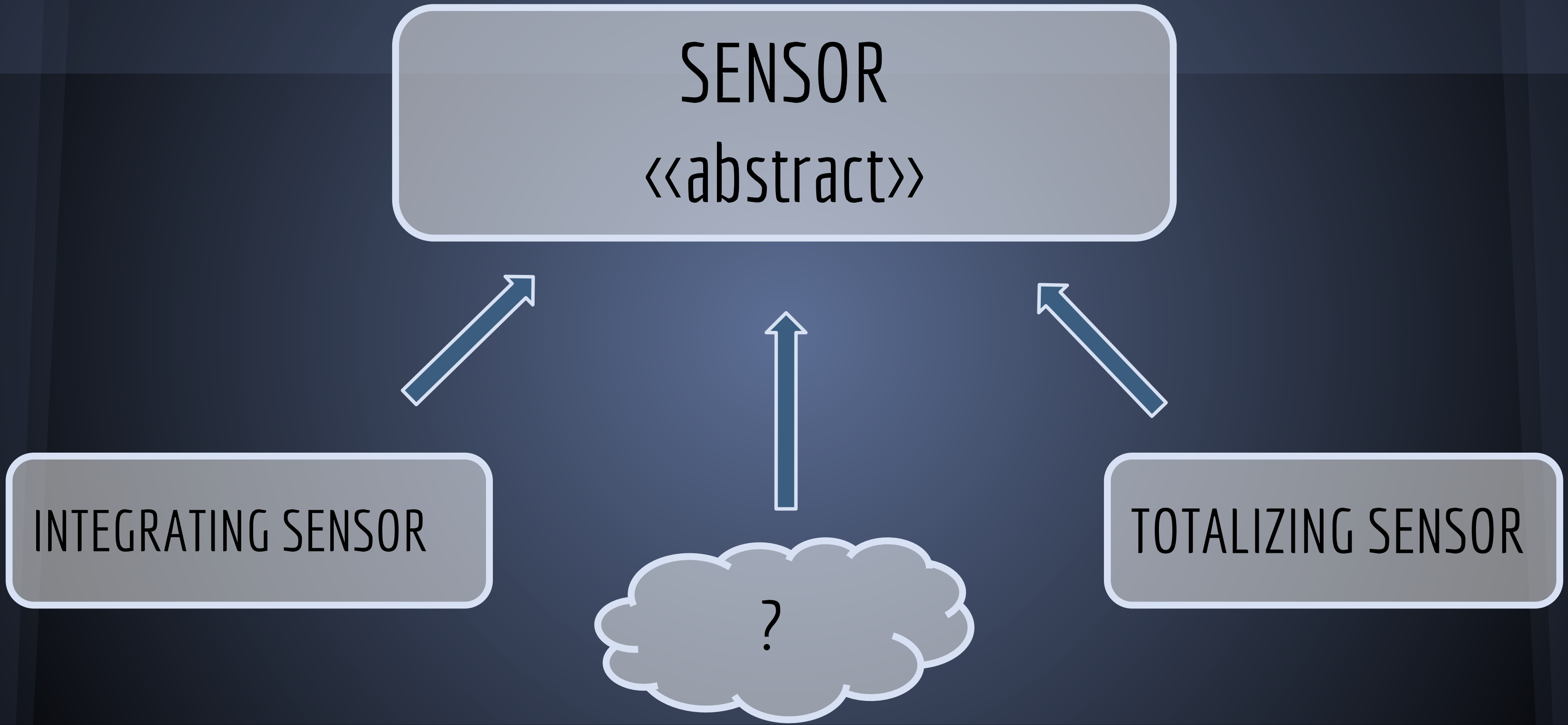
SUM CONSUMED()

INTEGRATING SENSOR

SUM CONSUMED()

TOTALIZING SENSOR

SUM CONSUMED()



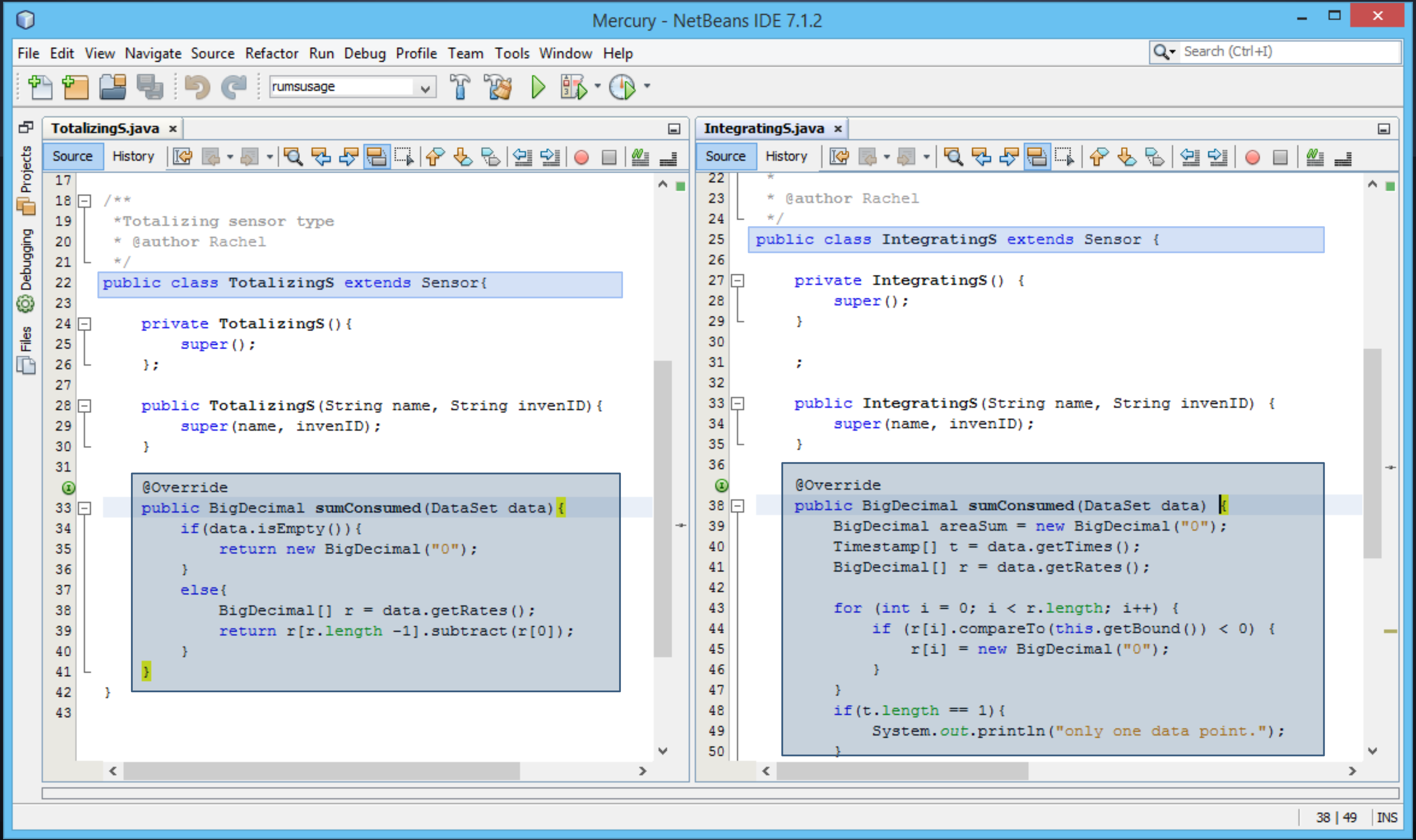
SENSOR

«abstract»

INTEGRATING SENSOR

?

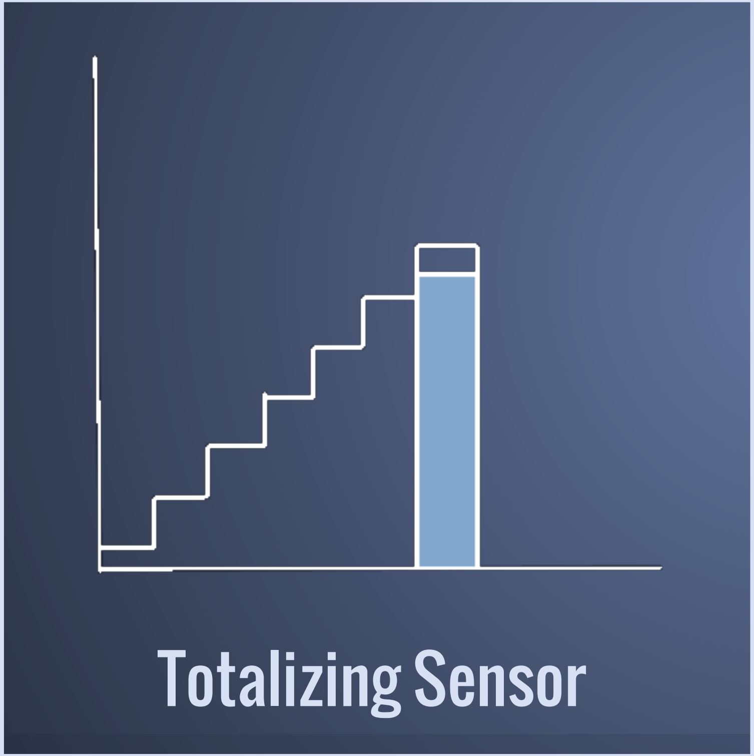
TOTALIZING SENSOR


Mercury - NetBeans IDE 7.1.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
rumsusage
Projects
Debugging
Files
TotalizingS.java
IntegratingS.java
Source
History
38 | 49
INS

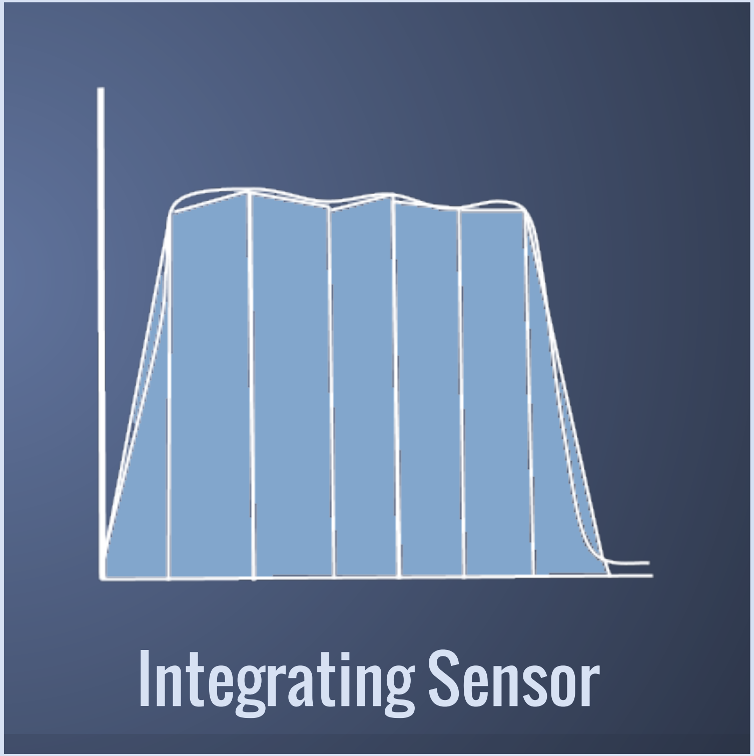
```java
/**
 *Totalizing sensor type
 * @author Rachel
 */
public class TotalizingS extends Sensor{

    private TotalizingS(){
        super();
    };

    public TotalizingS(String name, String invenID){
        super(name, invenID);
    }

    @Override
    public BigDecimal sumConsumed(DataSet data){
        if(data.isEmpty()){
            return new BigDecimal("0");
        }
        else{
            BigDecimal[] r = data.getRates();
            return r[r.length -1].subtract(r[0]);
        }
    }
}
```

```java
 *
 * @author Rachel
 */
public class IntegratingS extends Sensor {

    private IntegratingS() {
        super();
    }

    ;

    public IntegratingS(String name, String invenID) {
        super(name, invenID);
    }

    @Override
    public BigDecimal sumConsumed(DataSet data) {
        BigDecimal areaSum = new BigDecimal("0");
        Timestamp[] t = data.getTimes();
        BigDecimal[] r = data.getRates();

        for (int i = 0; i < r.length; i++) {
            if (r[i].compareTo(this.getBound()) < 0) {
                r[i] = new BigDecimal("0");
            }
        }
        if(t.length == 1){
            System.out.println("only one data point.");
        }
```

sumConsumed() Algorithms



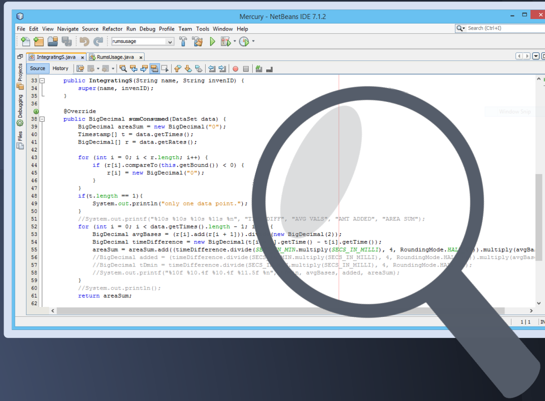
Totalizing Sensor



Integrating Sensor

Lab Work Objectives

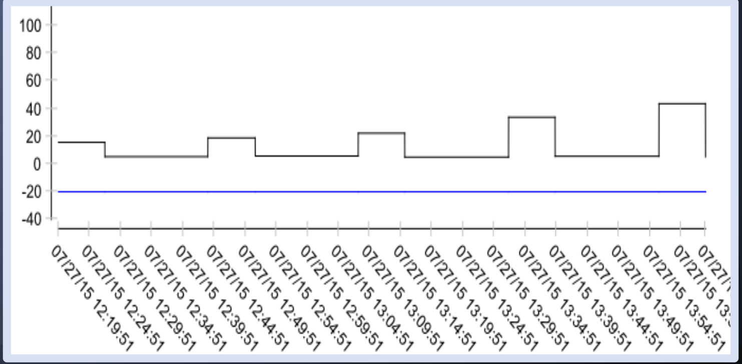
- debug software
- map voltage to flow rates using a regression series
- test GeH_4 flow at different rates
- test Ge deposition properties
  - stress using flexus


Mercury - NetBeans IDE 7.1.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
rumsusage
IntegratingS.java
RumsUsage.java
Source
History
Projects
Debugging
Files
1 | 1
INS

```
    public IntegratingS(String name, String invenID) {
        super(name, invenID);
    }

    @Override
    public BigDecimal sumConsumed(DataSet data) {
        BigDecimal areaSum = new BigDecimal("0");
        Timestamp[] t = data.getTimes();
        BigDecimal[] r = data.getRates();

        for (int i = 0; i < r.length; i++) {
            if (r[i].compareTo(this.getBound()) < 0) {
                r[i] = new BigDecimal("0");
            }
        }
        if(t.length == 1){
            System.out.println("only one data point.");
        }
        //System.out.printf("%10s %10s %10s %11s %n", "TI[illegible]DIFF", "AVG VALS", "AMT ADDED", "AREA SUM");
        for (int i = 0; i < data.getTimes().length - 1; i[illegible] {
            BigDecimal avgBases = (r[i].add(r[i + 1])).di[illegible](new BigDecimal(2));
            BigDecimal timeDifference = new BigDecimal(t[i[illegible]].getTime() - t[i].getTime());
            areaSum = areaSum.add((timeDifference.divide(SE[illegible]N_MIN.multiply(SECS_IN_MILLI), 4, RoundingMode.HAL[illegible])).multiply(avgBa
            //BigDecimal added = (timeDifference.divide(SECS[illegible]MIN.multiply(SECS_IN_MILLI), 4, RoundingMode.HAL[illegible])).multiply(avgBas
            //BigDecimal tDmin = timeDifference.divide(SECS_I[illegible].multiply(SECS_IN_MILLI), 4, RoundingMode.HA[illegible]);
            //System.out.printf("%10f %10.4f %10.4f %11.5f %n", [illegible]n, avgBases, added, areaSum);
        }
        //System.out.println();
        return areaSum;
```

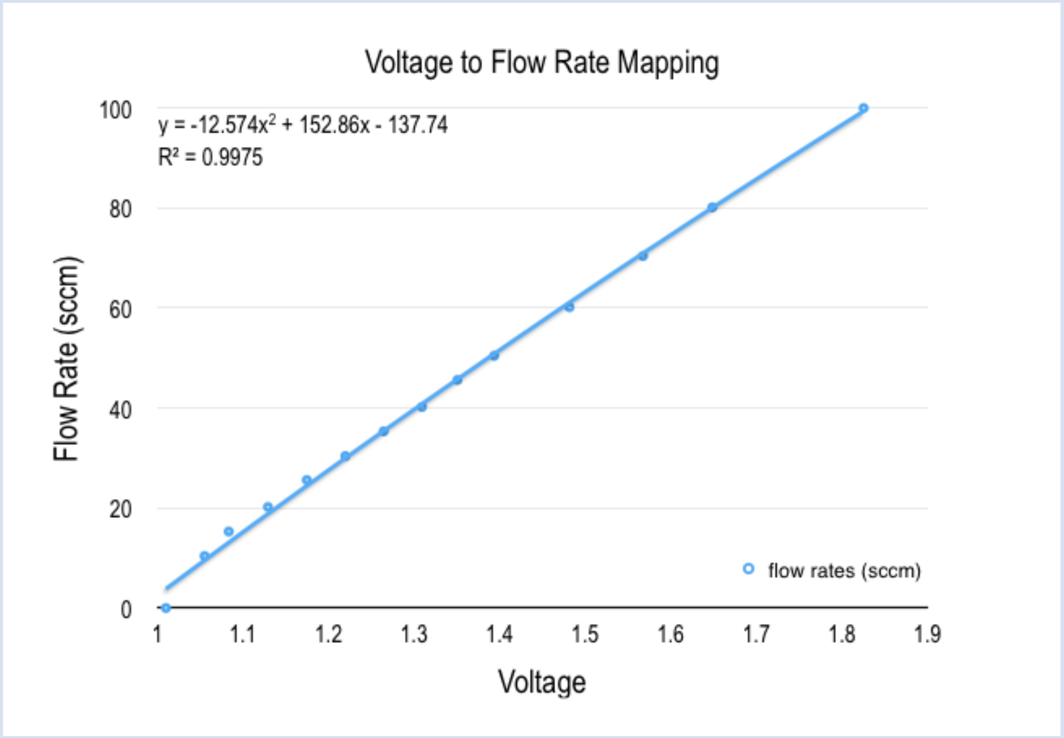
First Run on the AMATEPI

- ran at 1, 5, 10, 20, 30, 40, 50, 75, 100 sccm
- program crashed
  - incomplete parameters
- RUMS nano charting was inaccurate



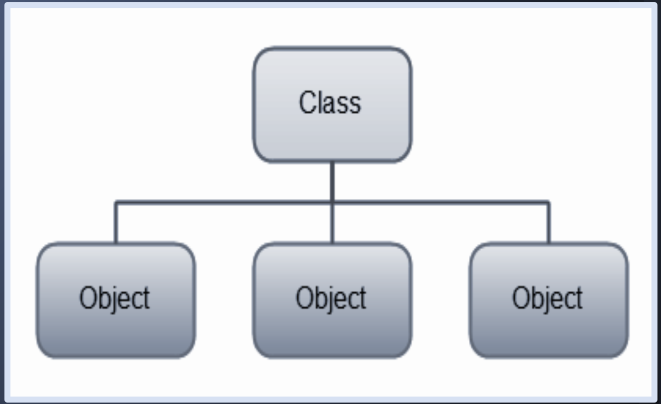
Improving Data Accuracy

- sensors calibrated
- mapped voltage to actual flow rates rather than programmed rates
- 30.4 sccm: now 29.99 rather than 27.5
- r^2 value of .998 for flow rates up to 100 sccm



What I learned In the Process

- dynamic programs
- choice of variable and list types
- communication
- failure during testing



Acknowledgements

Thank you…

- Olek Proskurowski and Richelieu Hemphill for mentoring me
- Marilyn, Jayss, and Cheryl for taking us to Semicon and helping me in the lab
- Bill Flounders for giving me the opportunity to intern this summer