Document Control

Approval

The Guidance Team and the customer shall approve this document.

Document Change Control

<table>
<thead>
<tr>
<th>Initial Release:</th>
<th>1.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current Release:</td>
<td>2.0</td>
</tr>
<tr>
<td>Indicator of Last Page in Document:</td>
<td>Q.E.D.</td>
</tr>
<tr>
<td>Date of Last Review:</td>
<td>04/13/2009</td>
</tr>
<tr>
<td>Date of Next Review:</td>
<td>NA</td>
</tr>
<tr>
<td>Target Date for Next Update:</td>
<td>NA</td>
</tr>
</tbody>
</table>

Distribution List

This following list of people shall receive a copy of this document every time a new version of this document becomes available:

Guidance Team Members:
Dr. Steven Roach
Ms. Evelyn Torres

Customer:
Dr. Craig Tweedie
Department of Biology
413 Biology Building
c tweedie@utep.edu
(915)-747-8448
http://faculty.utep.edu/Default.aspx?alias=faculty.utep.edu/ctweedie

Software Team Members:
Christopher Cuellar
Jon Mcelyea
Nicolle Whitman
Luis Fregoso
Manuel Corona
Brenda Medina

Change Summary

The following table details changes made between versions of this document

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Modifier</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>03/27/2009</td>
<td>Team</td>
<td>Initial Document Release</td>
</tr>
<tr>
<td>2.0</td>
<td>04/13/2009</td>
<td>N. Whitman, L. Fregoso, &amp; B. Medina</td>
<td>Complete document restructure, and correction of most sections.</td>
</tr>
</tbody>
</table>
Table of Contents

DOCUMENT CONTROL ........................................................................................................ II

APPROVAL ...................................................................................................................... II
DOCUMENT CHANGE CONTROL .................................................................................. II
DISTRIBUTION LIST ........................................................................................................ II
CHANGE SUMMARY ....................................................................................................... II

1 INTRODUCTION .......................................................................................................... 1

1.1 PURPOSE AND INTENDED AUDIENCE ................................................................. 1
1.2 SCOPE OF PRODUCT ............................................................................................. 1
1.3 REFERENCES ........................................................................................................... 1
1.4 DEFINITIONS, ACRONYMS, AND ABBREVIATIONS ........................................... 1
   1.1.1 Definitions ........................................................................................................ 1
   1.4.1 Acronyms and Abbreviations ......................................................................... 1
1.5 OVERVIEW ................................................................................................................ 2

2 DECOMPOSITION DESCRIPTION ............................................................................. 3

2.1 SYSTEM COLLABORATION DIAGRAM ............................................................... 3
2.2 SUBSYSTEM AND COMPONENT DESCRIPTION ................................................... 5

3 DETAILED DESCRIPTION OF ANALYSIS COMPONENT ......................................... 7

3.1 COMPONENT DESCRIPTION .................................................................................. 7
3.2 CLASS DESCRIPTION: OUTSIDE ENTITY ............................................................ 7
   3.2.1 Contract: Outside Entity 1 .............................................................................. 7
3.3 CLASS DESCRIPTION: SEARCH SUBSYSTEM ..................................................... 7
   3.3.1 Contract: Search Subsystem 1 ....................................................................... 7
3.4 CLASS DESCRIPTION: DATA ANALYSIS ............................................................ 8
   3.4.1 Contract: Data Analysis 1 ............................................................................. 8
   3.4.2 Contract: Data Analysis 2 ............................................................................. 9
3.5 CLASS DESCRIPTION: DISPLAY .......................................................................... 9
3.6 CLASS DESCRIPTION: WEB SERVICE ACCESSOR .............................................. 10
   3.6.1 Contract: Web Service Accessor 1 ............................................................... 10
   3.6.2 Contract: Web Service Accessor 2 ............................................................... 10
3.7 CLASS DESCRIPTION: EXECUTE REDUCTION QUERY .................................. 10
   3.7.1 Contract: Execute Query 1 ........................................................................... 11
   3.7.2 Contract: Execute Query 2 ........................................................................... 11
3.8 CLASS DESCRIPTION: REDUCTION QUERY INTERPRETER ................................ 11
   3.8.1 Contract: Execution Parser 1 ........................................................................ 12
   3.8.2 Contract: Execution Parser 2 ........................................................................ 12
   3.8.3 Contract: Execution Parser 3 ........................................................................ 12
   3.8.4 Contract: Execution Parser 4 ........................................................................ 13
3.9 CLASS DESCRIPTION: APPLY BOOLEAN EXPRESSION .................................. 13
   3.9.1 Contract: Apply Boolean 1 .......................................................................... 13
3.10 CLASS DESCRIPTION: APPLY TIME CONSTRAINTS ......................................... 13
   3.10.1 Contract: Apply Time 1 .............................................................................. 14
3.11 CLASS DESCRIPTION: ANALYSIS LANGUAGE ............................................... 14
   3.11.1 Contract: Analysis Language 1 .................................................................. 14
3.12 CLASS DESCRIPTION: R .................................................................................... 16
   3.12.1 Contract: Analysis Language 1 .................................................................. 17
3.13 CLASS DESCRIPTION: REDUCTION TECHNIQUE .......................................... 17
   3.13.1 Contract: Reduction Technique 1 .............................................................. 17
   3.13.2 Contract: Reduction Technique 2 .............................................................. 17
3.14 CLASS DESCRIPTION: AVERAGE ........................................................................ 18
3.14.1 Contract: Reduction Technique 1 .................................................. 18
3.14.2 Contract: Reduction Technique 2 .................................................. 18
3.15 CLASS DESCRIPTION: MEDIAN ....................................................... 18
    3.15.1 Contract: Reduction Technique 1 ............................................. 18
    3.15.2 Contract: Reduction Technique 2 ............................................. 18
3.16 CLASS DESCRIPTION: MODE .......................................................... 18
    3.16.1 Contract: Reduction Technique 1 ............................................. 19
    3.16.2 Contract: Reduction Technique 2 ............................................. 19
3.17 CLASS DESCRIPTION: KURTOSIS .................................................... 19
    3.17.1 Contract: Reduction Technique 1 ............................................. 19
    3.17.2 Contract: Reduction Technique 2 ............................................. 19
3.18 CLASS DESCRIPTION: SKEW ........................................................... 19
    3.18.1 Contract: Reduction Technique 1 ............................................. 19
    3.18.2 Contract: Reduction Technique 2 ............................................. 19
3.19 CLASS DESCRIPTION: COVARIANCE .............................................. 19
    3.19.1 Contract: Reduction Technique 1 ............................................. 20
    3.19.2 Contract: Reduction Technique 2 ............................................. 20
3.20 CLASS DESCRIPTION: HISTOGRAM PLOT ....................................... 20
    3.20.1 Contract: Reduction Technique 1 ............................................. 20
    3.20.2 Contract: Reduction Technique 2 ............................................. 20
3.21 CLASS DESCRIPTION: VARIABLE CORRELATION ............................. 20
    3.21.1 Contract: Reduction Technique 1 ............................................. 20
    3.21.2 Contract: Reduction Technique 2 ............................................. 21
3.22 CLASS DESCRIPTION: HYPOTHESIS TESTING ................................. 21
    3.22.1 Contract: Reduction Technique 1 ............................................. 21
    3.22.2 Contract: Reduction Technique 2 ............................................. 21
3.23 CLASS DESCRIPTION: MAXIMUM .................................................... 21
    3.23.1 Contract: Reduction Technique 1 ............................................. 21
    3.23.2 Contract: Reduction Technique 2 ............................................. 21
3.24 CLASS DESCRIPTION: MINIMUM .................................................... 21
    3.24.1 Contract: Reduction Technique 1 ............................................. 22
    3.24.2 Contract: Reduction Technique 2 ............................................. 22
3.25 CLASS DESCRIPTION: STANDARD DEVIATION .................................. 22
    3.25.1 Contract: Reduction Technique 1 ............................................. 22
    3.25.2 Contract: Reduction Technique 2 ............................................. 22
3.26 CLASS DESCRIPTION: SUM ............................................................ 22
    3.26.1 Contract: Reduction Technique 1 ............................................. 22
    3.26.2 Contract: Reduction Technique 2 ............................................. 22
3.27 CLASS DESCRIPTION: BAR GRAPH ............................................... 23
    3.27.1 Contract: Reduction Technique 1 ............................................. 23
    3.27.2 Contract: Reduction Technique 2 ............................................. 23
3.28 CLASS DESCRIPTION: SCATTER PLOT .............................................. 23
    3.28.1 Contract: Reduction Technique 1 ............................................. 23
    3.28.2 Contract: Reduction Technique 2 ............................................. 23
3.29 CLASS DESCRIPTION: LINE GRAPH ............................................... 23
    3.29.1 Contract: Reduction Technique 1 ............................................. 23
    3.29.2 Contract: Reduction Technique 2 ............................................. 24
3.30 CLASS DESCRIPTION: DENSITY CHART .......................................... 24
    3.30.1 Contract: Reduction Technique 1 ............................................. 24
    3.30.2 Contract: Reduction Technique 2 ............................................. 24
3.31 CLASS DESCRIPTION: BOX PLOT .................................................... 24
    3.31.1 Contract: Reduction Technique 1 ............................................. 24
    3.31.2 Contract: Reduction Technique 2 ............................................. 24
3.32 CLASS DESCRIPTION: PIE CHART ................................................. 24
    3.32.1 Contract: Reduction Technique 1 ............................................. 25
    3.32.2 Contract: Reduction Technique 2 ............................................. 25
3.33  CLASS DESCRIPTION: DECISION MAKING ................................................................. 25
  3.33.1  Contract: Reduction Technique 1 ................................................................. 25
  3.33.2  Contract: Reduction Technique 2 ................................................................. 25

3.34  CLASS DESCRIPTION: CLASSIFICATION .............................................................. 25
  3.34.1  Contract: Reduction Technique 1 ................................................................. 25
  3.34.2  Contract: Reduction Technique 2 ................................................................. 26

3.35  CLASS DESCRIPTION: STRIP PLOT ..................................................................... 26
  3.35.1  Contract: Reduction Technique 1 ................................................................. 26
  3.35.2  Contract: Reduction Technique 2 ................................................................. 26

3.36  CLASS DESCRIPTION: REGRESSION ................................................................. 26
  3.36.1  Contract: Reduction Technique 1 ................................................................. 26
  3.36.2  Contract: Reduction Technique 2 ................................................................. 26
1 Introduction

1.1 Purpose and Intended Audience
The Software Design Document (SDD) establishes the software design of the Automatic Weather Station Database (AWSD) System. This document is intended to aid Caffeine Inc. in the development and implementation of the AWSD System and it is intended to serve as a technical tool and reference for the teams of the software engineering class, spring 2009.

1.2 Scope of Product
Dr. Craig Tweedie of the Department of Biological Sciences at the University of Texas at El Paso (UTEP) is interested in modeling and understanding climate patterns, more specifically, he is interested in determining human impact on climate. To this end, Dr. Tweedie is in the process of placing weather stations at various sites including arctic regions and areas in the desert southwest. The current challenges faced by scientists such as Dr. Tweedie are:

- As more data becomes available, how can data be collected and distributed to other scientists who are interested in obtaining the data?
- Can the data be trusted to be reliable?
- Scientists will require software support to reduce, manipulate, and interpret data.

The AWSD system is the proposed software system to provide support for the collection, assessment, distribution, reduction, and manipulation of weather and climate data. The AWSD system is intended for research purposes and will be employed for different research needs.

1.3 References

1.4 Definitions, Acronyms, and Abbreviations

1.4.1 Definitions

<table>
<thead>
<tr>
<th>TERM</th>
<th>DEFINITION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Design Component</td>
<td>A subsystem</td>
</tr>
<tr>
<td>Design Entity</td>
<td>A subsystem or class</td>
</tr>
</tbody>
</table>

1.4.1 Acronyms and Abbreviations
No abbreviations were used in this document.

### 1.5 Overview

This document is divided into three main sections. The first section covers the purpose of the document, the intended audience, references used, and the definitions, acronyms and abbreviations used in the document. The second section provides an overview of the subsystems which will be integrated to create the AWSD system. The third section covers the design and description of the subsystem that our team was assigned and how it works with the rest of the system.

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AI</td>
<td>Artificial Intelligence</td>
</tr>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>AWSD</td>
<td>Automatic Weather Station Database</td>
</tr>
<tr>
<td>CRC</td>
<td>Class, Responsibilities, and Collaborations</td>
</tr>
<tr>
<td>SDD</td>
<td>Software Design Document</td>
</tr>
<tr>
<td>SRS</td>
<td>Software Requirements Specification</td>
</tr>
<tr>
<td>UI</td>
<td>User Interface</td>
</tr>
<tr>
<td>UTEP</td>
<td>University of Texas at El Paso</td>
</tr>
</tbody>
</table>
2 Decomposition Description

The descriptions and diagrams provided in this document will aid Caffeine Inc. and the teams from the software engineering class by:

- Defining an interface to classes which will aid developers and maintainers in determining how to use the existing classes.
- Listing class contracts which enable designers and maintainers to understand how the system functions and it would enable them to trace requirements to design entities.
- Providing a system diagram which illustrates how the system components or subsystems interact with each other.

2.1 System Collaboration Diagram

The AWSD system was broken down into the following subsystems:

- System Support to be developed and implemented by Coffee Break
- Data Provider to be developed and implemented by Late Night Alumni
- Search to be developed and implemented by Red Storm
- Analysis to be developed and implemented by Caffeine Inc.
- Database to be developed and implemented by Cat6

Figure 2.1 illustrates the subsystems of the AWSD system and their interaction. Figure 2.2 depicts the CRC diagram and the class hierarchy for the Analysis subsystem.

Figure 2.1: System diagram that illustrates the collaboration of the subsystems.

The following is a description of each of the subsystems illustrated in Figure 2.1:

- System Support subsystem is responsible for the following:
  - The user interface (UI) for the administrator
  - Collecting data and generating administrative reports.
  - Editing data.
  - The artificial intelligence for flagging.
Software Design Document

- The UI for flagging.
- Data Provider subsystem is responsible for the following:
  - Configuring weather stations and sensors.
  - Uploading data which includes concepts like parsing, matching, flagging, delimiters, and order.
- Search subsystem is responsible for the following:
  - The simple search functionality
  - The advanced search functionality
  - The download functionality
  - The save-queries functionality
- Analysis subsystem is responsible for the following:
  - Data reduction through R.
  - Data analysis through R
  - Data plotting through R.
  - Displaying the results
  - Allowing the user to download the results
- Database subsystem is responsible for the following:
  - The database API through postgreSQL server
  - Unit conversions
  - Executing queries
  - Update notifications

Section 3 will present a description of the relationships among the entities in the Analysis subsystem; very few design, development, and implementation details are known for the other subsystems thus far.
2.2 Subsystem and Component Descriptions

The logic or 'intelligence' behind the Analysis subsystem will be accessible through a web service. The Search team will access the Analysis subsystem through the Display entity while the Upload and Data Provider teams will access the Analysis subsystem through the Web Service Accessor (see Figure 2.2); this is due to the fact that only the Search team requires the Analysis subsystem to interact with the user. The Data Analysis entity will be responsible for sending the input to the appropriate class and returning the results to the Web Service Accessor. Reduction Query Interpreter is responsible for calling the appropriate class that applies any Boolean expressions entered by the user, applies any time constraints specified by the user, and submitting the data to the correct reduction technique; the reduction technique will parse the parameters into the order in which the Analysis Language is expecting them in.

The Reduction Query will follow the subsequent structure:

Reduction Query := Function [Boolean Expression] [Time Constraints] {Parameters}
Where Function could be sum, median, line graph, or any of the available functions.

Note the following:
- The square brackets and the curly brackets are part of the reduction query.
- The parameters will be function dependent.
- If no Boolean expression or time constraints are specified by the user, an empty pair of square brackets is needed.
- If no parameters are needed for a specific function, as is the case of sum, curly brackets cannot be present.

The following is still in discussion among the team:
- The parameters needed for each type of function
- The order in which the parameters must be specified within the reduction query
- Default values if no parameters are specified by the user, such as graph title or axis titles.
- The structure of Boolean Expression and Time Constraints
3 Detailed Description of Analysis Component

The detailed design choices presented in this section will aid the designers, programmers, and maintainers by:

- Allowing the delegation of implementation to the programmers in Caffeine Inc.
- Simplifying the creation of test cases; test cases can be derived from the pre and post conditions.
- Allowing designers of Caffeine Inc. to detect flaws in the design at an early stage.

3.1 Component Description

The Analysis Subsystem is responsible for data reduction, data analysis, data plotting, result displays, and result downloads. The class contracts and the class hierarchies among the classes of the Analysis Subsystem are illustrated in Figure 2.2. The following is a list of the classes that will be implemented as the Analysis Subsystem: Display, Web Service Accessor, Data Analysis, Execute Reduction Query, Reduction Query Interpreter, Apply Boolean Expression, and Apply Time Constraints. In addition, there are two abstract classes: Reduction Technique and Analysis Language. R is a concrete class that is a subclass of Analysis Language. The following is a list of concrete classes that are subclasses of the Reduction Technique abstract class: Average, Variable Correlation, Histogram Plot, Strip Plot, Maximum, Median, Skew, Classification, Box Plot, Minimum, Mode, Covariance, Decision Making, Pie Chart, Sum, Kurtosis, Hypothesis testing, Regression, Density Chart, Standard Deviation, Scatter Plot, Line Graph, and Bra Graph.

3.2 Class Description: Outside Entity

Outside Class: Outside Entity
1. Send information to Web Service Accessor
   a. Send data object to be analyzed
   b. Send a string containing the analysis to be performed

Description: This class refers to any team or other source that might be interested in calling the Analysis Subsystem, but is outside the Analysis subsystem boundaries.

Private Responsibilities:
Receive the results from Web Service Accessor WSA (2)

3.2.1 Contract: Outside Entity 1

Contract Identifier: OE1
Contract Name: Outside Entity 1

The outside entity will follow the OE1 contract by providing the input data in a predefined php data object, as specified by the database team, and a reduction query string in the structure defined in this document.

3.3 Class Description: Search Subsystem

Outside Class: Search Subsystem
1. Send information to Display
   a. Send data object to be analyzed
   b. Send a string containing the user’s saved analyses

Description: This class refers to the Search Subsystem that will send control to the Analysis subsystem which will interact with the user.

Private Responsibilities: None

3.3.1 Contract: Search Subsystem 1

Contract Identifier: SS1
Contract Name: Search Subsystem 1
The outside entity will follow the OE1 contract by providing the input data in a predefined php data object, as specified by the database team.

### 3.4 Class Description: Data Analysis

**Concrete Class: Data Analysis**

1. Passes information to Execute Query
   - a. Passes active data set
   - b. Passes analysis string

2. Return the result to Web Service Accessor
   - a. Passes result

**Description:** This class will be the interface between the web service and the classes that perform the reduction and analysis.

**Private Responsibilities:**

- Receive active dataset to be analyzed from Web Service Accessor WSA (1)
- Receive the results from Execute Reduction Query EQ (2)

### 3.4.1 Contract: Data Analysis 1

**Contract Identifier:** DA1

**Contract Name:** Data Analysis 1

The Data Analysis class will follow the DA1 contract by sending the appropriate data to the appropriate class to get the data reduced and submitted to R.

**Signature:** dataAnalysisTechniques(self, data, sensors, reductionType, parameters)

**Method Name:** dataAnalysisTechniques

**Type of return value:** reductionTechnique object

**Type of input parameters:** self, data is a list, sensors is a list, reductionType is a string, parameters is a list

**Type of output parameters:** none

**Description of input and output parameters:**

- Self is the class that the method is contained in. This allows the method to call variables that are contained within the class.
- Data is a list of the necessary data to perform the reduction technique passed.
- Sensors is a list of sensors which will be utilized for the naming schemes in the reduction technique.
- ReductionType is the type of reduction that will be performed.
- Parameters is a list of the parameters which will be needed for some of the reduction techniques.

**Purpose:**

This method returns a reduction type object based on the reductionType passed.

**Pre-Conditions:**

- The data specified in the initialization must be reducible by the reduction technique that is specified in the initialization.
- The data specified must not be null.
- The data specified must have been initialized and have at least one piece of data in it.

**Post-Conditions:**

- Does not delete, append or modify in any way type or form the data used to produce the specific reduction technique.
- Returns a reduction type object based on the reductionType passed

**Signature:** returnResults (self, OperationObjects)

**Method Name:** returnResults

**Type of return value:** Returns a list of the values returned by the reduction techniques.

**Type of input parameters:** self, List of ReductionTechnique- OperationObjects

**Type of output parameters:** none

**Description of input and output parameters:**
Self is the class that the method is contained in. This allows the method to call variables that are contained within the class.

OperationObjects is a list of ReductionTechniques that is iterated to perform the analysis.

**Purpose:**
This method iterates a list of ReductionTechniques and calls their individual analysis methods and returns a list of data that they returned.

**Pre-Conditions:**
The list of OperationObjects, which are ReductionTechniques, specified must not be null.

**Post-Conditions:**
Does not delete, append or modify in any way type or form the data used to produce the specific reduction technique.

### 3.4.2 Contract: Data Analysis 2

**Contract Identifier:** DA2  
**Contract Name:** Data Analysis 2

The Data Analysis class will follow the DA contract by returning the accurate reduced data. For the protocols of the methods involved see section 3.4.1.

### 3.5 Class Description: Display

**Concrete Class:** Display  
**Private Responsibilities:**
- Display active data set
- Display data format options for reduced data
- Display reduction options
- Communicate with Web Service Accessor to receive results
- Display results

**Description:** This class is responsible for interacting with the user and collecting the user’s options to be sent through the web service, it will generate the reduction query string based on the user’s options, display data, and allow the user to download the results.

**Private Responsibilities:** None specified

The following is the method protocol for the Display class:

**Signature:** Display(Data to be reduced, Past reduction string)  
**Method Name:** Display  
**Type of return value:** none  
**Type of input parameters:** Data, a string specifying past reductions a user has saved

**Type of output parameters:** none

**Description of input and output parameters:**
Data is the set of data on which the reduction technique will be applied to.
Past reduction string is a string that specifies all the reductions that the user has saved

**Purpose:**
Display the user interface so that the user can select the types of reduction techniques to apply and on which part of the data to apply it.

**Pre-Conditions:**
Data is not null.
If no saved reductions exist an empty string should be passed.

**Post-Conditions:**
Does not delete, append or modify in any way type or form the data used to produce the specific reduction technique.
GUI is displayed.
3.6 Class Description: Web Service Accessor

Concrete Class: Web Service Accessor
1. Sends information to Data Analysis
   a. Sends active dataset
   b. Sends analysis string
2. Sends information to Outside Entity
   a. Sends the results to Outside Entity

Description: This class will implement a web service to provide data reduction and data analysis through R.

Private Responsibilities:
Knows what to send to Data Analysis
Receive results from Data Analysis
Knows what to send to Outside Entity

3.6.1 Contract: Web Service Accessor 1

Contract Identifier: WSA1
Contract Name: Web Service Accessor 1

The Web Service Accessor will follow WSA1 contract by submitting the correct data and in the correct order to the Data Analysis class for it to provide the data reduction and analysis service using R.

Signature: PerformAnalysis(Data to be reduced, Analysis string)
Method Name: PerformAnalysis
Type of return value: image or string
Type of input parameters: Data, Analysis string
Type of output parameters: none
Description of input and output parameters:
Data is the set of data on which the reduction technique will be applied to.
Analysis string is a string specifying the reduction techniques that will be applied. It includes the reduction name, Boolean expression if it applies, time constraints and any extra parameters needed for the reduction, graph labels for example.
Purpose:
Web service that processes and returns the results from a reduction technique which is specified in the Analysis string.
Pre-Conditions:
Data is not null.
Analysis string is in the correct format as specified in section 2.2
Post-Conditions:
Returns the reduction results.

3.6.2 Contract: Web Service Accessor 2

Contract Identifier: WSA2
Contract Name: Web Service Accessor 2

The Web Service Accessor will follow WSA2 contract by returning the accurate results according to the data and reduction query input. For the protocol of the methods involved see section 3.6.1.

3.7 Class Description: Execute Reduction Query

Concrete Class: Execute Reduction Query
1. Creates an instance of Reduction Query Interpreter
   a. Sends active dataset
   b. Sends analysis string
2. Sends information to Data Analysis
   a. Sends the results to Data Analysis
Description: This class is responsible for providing an interface between the Data Analysis and the Reduction Query Interpreter class.

Private Responsibilities:
- Knows what type of Reduction Query Interpreter needs to be created
- Knows what to send to Data Analysis
- Receives analysis string from Data Analysis
- Receives active data set from Data Analysis
- Receives results from Reduction Query Interpreter

3.7.1 Contract: Execute Query 1

Contract Identifier: EQ1
Contract Name: Execute Query 1

The Execute Reduction Query class will follow the EQ1 contract by submitting the data and the reduction query string to the appropriate classes for it to be reduced and analyzed, in the correct order.

Signature: ExecuteReductionQuery(Data to be reduced, Analysis string)
Method Name: ExecuteReductionQuery
Type of return value: image or string
Type of input parameters: Data, Analysis string
Type of output parameters: none

Description of input and output parameters:
- Data is the set of data on which the reduction technique will be applied to.
- Analysis string is a string specifying the reduction techniques that will be applied. It includes the reduction name, Boolean expression if it applies, time constraints and any extra parameters needed for the reduction, graph labels for example.

Purpose:
- This class aids the Web service in reducing the data and submitting it to R, and then providing the results.

Pre-Conditions:
- Data is not null.
- Analysis string is in the correct format as specified in section 2.2.

Post-Conditions:
- Returns the reduction results.

3.7.2 Contract: Execute Query 2

Contract Identifier: EQ2
Contract Name: Execute Query 2

The Execute Reduction Query class will follow the EQ2 by returning the results of the data after applying the reductions and functions. For the protocol of the methods involved see section 3.7.1.

3.8 Class Description: Reduction Query Interpreter

Concrete Class: Reduction Query Interpreter
1. Send information to Execute Reduction Query
   a. Sends the results
2. Creates instance(s) of Apply Boolean Expression
   a. Passes Boolean expression string
3. Creates instance(s) of Apply Time Constraints
   a. Passes time-constraints string
4. Creates instance(s) of Reduction Technique
   a. Passes reduced active data set

Description: This class is responsible for sequencing the following steps: applying Boolean expression to the data, applying time constrains to the data, executing the function specified and returning the results.

Private Responsibilities:
Knows the Apply Boolean expression
Knows the Apply Time Constraints
Knows the Reduction Technique
Knows what to pass to Apply Boolean Expression
Knows what to pass to Apply Time Constraints
Knows what to pass to Reduction Technique
Knows what to send to Execute Reduction Query
Receives analysis string from Execute Reduction Query
Receives active data set from Execute Reduction Query
Receives reduced active data set from Apply Boolean Expression
Receives reduced active data set from Apply Time Constraints
Receives result from Reduction Technique

3.8.1 Contract: Execution Parser 1

Contract Identifier: EQ1
Contract Name: Execution Parser 1

The Reduction Query Interpreter class will follow the EQ1 contract by returning the accurate results as specified by the data and reduction query input.

Signature: performReductionAndApplyFunction(Data to be reduced, Analysis string)
Method Name: performReductionAndApplyFunction
Type of return value: image or string
Type of input parameters: Data, Analysis string
Type of output parameters: none
Description of input and output parameters:
Data is the set of data on which the reduction technique will be applied to.
Analysis string is a string specifying the reduction techniques that will be applied. It includes the reduction name, Boolean expression if it applies, time constraints and any extra parameters needed for the reduction, graph labels for example.
Purpose:
This class aids the Web service in reducing the data and submitting it to R, and then providing the results. It will interpret the analysis string, reduce the data, submit the data to R through the interface provided, and then return the results.
Pre-Conditions:
Data is not null.
Analysis string is in the correct format as specified in section 2.2.
Post-Conditions:
Returns the reduction results.

3.8.2 Contract: Execution Parser 2

Contract Identifier: EQ2
Contract Name: Execution Parser 2

The Reduction Query Interpreter class will follow the EQ2 contract by submitting the data in the correct order to the class that interfaces with R. For a protocol of the methods involved see section 3.8.1.

3.8.3 Contract: Execution Parser 3

Contract Identifier: EQ3
Contract Name: Execution Parser 3

The Reduction Query Interpreter class will follow the EQ3 contract by submitting the data and a Boolean expression string in the correct order and format to the appropriate class so the data can be accurately reduced according to the Boolean expression. For a protocol of the methods involved see section 3.8.1.
3.8.4 Contract: Execution Parser 4
Contract Identifier: EQ4
Contract Name: Execution Parser 4

The Reduction Query Interpreter class will follow the EQ3 contract by submitting the data and a Time constraints string in the correct order and format to the appropriate class so the data can be accurately reduced according to the time constraints. For a protocol of the methods involved see section 3.8.1.

3.9 Class Description: Apply Boolean Expression

Concrete Class: Apply Boolean Expression
1. Send reduced active data set to Reduction Query Interpreter
   a. Passes reduced active data set

Description: This class is responsible for reducing the data by applying the specified Boolean expression, task which involves parsing the Boolean expression, and returning the reduced data set.

Private Responsibilities:
Provide functions for Reduction Query Interpreter to apply a Boolean expression
Parses the Boolean expression string
Apply a Boolean expression to the active data set
Receives analysis string from Reduction Query Interpreter

3.9.1 Contract: Apply Boolean 1
Contract Identifier: AP1
Contract Name: Apply Boolean 1

The Apply Boolean Expression class will follow the AP1 contract by reducing and returning the data according to the specified Boolean expression given.

Signature: ApplyBooleanExpression(Data to be reduced, Boolean Expression string)
Method Name: ApplyBooleanExpression
Type of return value: List
Type of input parameters: Data, Boolean Expression string
Type of output parameters: none

Description of input and output parameters:
Data is the set of data on which the reduction technique will be applied to.
Boolean Expression string is a string specifying the Boolean expression to be applied to the data.

Purpose:
This class aids the Web service in reducing the data, and then providing the results.

Pre-Conditions:
Data is not null.
Boolean expression does not have a syntax error.

Post-Conditions:
Returns the reduced data

3.10 Class Description: Apply Time Constraints

Concrete Class: Apply Time Constraints
1. Send reduced active data set to Reduction Query Interpreter
   a. Passes reduced active data set

Description: This class is responsible for reducing the data by applying the data constraints specified, task that involves parsing the time constraints, and returning the reduced data set.

Private Responsibilities:
Provide functions for Execution Parser to apply a time reduction equation
Parses Time constraints string
Apply a time reduction equation to the active data set
Receives analysis string from Execution Parser

3.10.1 Contract: Apply Time 1
Contract Identifier: AT1
Contract Name: Apply Time 1

The Apply Time Constraints class will follow the AT1 contract by reducing and returning the data according to the specified time constraints given.

Signature: ApplyTimeConstraints(Data to be reduced, Time Constraints string)
Method Name: ApplyTimeConstraints
Type of return value: List
Type of input parameters: Data, Time Constraints string
Type of output parameters: none
Description of input and output parameters:
Data is the set of data on which the reduction technique will be applied to.
Time Constraints string is a string specifying the Time Constraints to be applied to the data.
Purpose:
This class aids the Web service in reducing the data, and then providing the results.
Pre-Conditions:
Data is not null.
Time Constraints does not have a syntax error (structure of the time constraints string is still pending finalization)
Post-Conditions:
Returns the reduced data

3.11 Class Description: Analysis Language
Abstract Class: Analysis Language
Subclass: R
1. Sends information to Reduction Technique
   a. Sends reduced active data set
Description: This class is responsible for enforcing a protocol for any analysis language that might be used; currently, R is the only Analysis Language of interest.
Private Responsibilities:
Provide Abstract functions for Reduction Technique to use appropriate Analysis Language
Receives active data set from Reduction Technique

3.11.1 Contract: Analysis Language 1
Contract Identifier: AL1
Contract Name: Analysis Language 1

The analysis Language class will follow the AL1 contract by providing a method to apply a certain function to the data specified utilizing R, and return the results.

Signature: PieGraph(self, data = none)
Method Name: PieGraph
Type of return value: Returns a string of the file path.
Type of input parameters: self, data is a list
Type of output parameters: none
Description of input and output parameters:
Self is the class that the method is contained in. This allows the method to call variables that are contained within the class.

Data is the list of data that will be plotted with this method.

**Purpose:**
This method returns a file that contains a pieGraph plot that has been performed on the data that was passed through.

**Pre-Conditions:**
The data specified must not be null.
The data specified must have been initialized and have at least one piece of data in it.

**Post-Conditions:**
Returns a file that contains a pieGraph plot that has been performed on the data that was passed through.

**Signature:** scatterPlot(self, data = none)
**Method Name:** scatterPlot
**Type of return value:** Returns a string of the file path.
**Type of input parameters:** self, data is a list
**Type of output parameters:** none

**Description of input and output parameters:**
Self is the class that the method is contained in. This allows the method to call variables that are contained within the class.

Data is the list of data that will be plotted with this method.

**Purpose:**
This method returns a file that contains a scatterPlot that has been performed on the data that was passed through.

**Pre-Conditions:**
The data specified must not be null.
The data specified must have been initialized and have at least one piece of data in it.

**Post-Conditions:**
Returns a file that contains a scatterPlot that has been performed on the data that was passed through.

**Signature:** LineGraph(self, data = none)
**Method Name:** LineGraph
**Type of return value:** Returns a string of the file path.
**Type of input parameters:** self, data is a list
**Type of output parameters:** none

**Description of input and output parameters:**
Self is the class that the method is contained in. This allows the method to call variables that are contained within the class.

Data is the list of data that will be plotted with this method.

**Purpose:**
This method returns a file that contains a LineGraph that has been performed on the data that was passed through.

**Pre-Conditions:**
The data specified must not be null.
The data specified must have been initialized and have at least one piece of data in it.

**Post-Conditions:**
Returns a file that contains a LineGraph that has been performed on the data that was passed through.

**Signature:** mean(self, data = none)
**Method Name:** mean
**Type of return value:** Returns a string containing the average of the data passed.
**Type of input parameters:** self, data is a list
**Type of output parameters:** none

**Description of input and output parameters:**
Self is the class that the method is contained in. This allows the method to call variables that are contained within the class.
Data is the list of data that will be plotted with this method.

**Purpose:**
This method returns a string containing the average of the data that was passed through.

**Pre-Conditions:**
The data specified must not be null.
The data specified must have been initialized and have at least one piece of data in it.

**Post-Conditions:**
Returns a string containing the average of the data that was passed through.

**Signature:** max(self, data = none)

**Method Name:** max

**Type of return value:** Returns a string of containing the average of the data passed.

**Type of input parameters:** self, data is a list

**Type of output parameters:** none

**Description of input and output parameters:**
Self is the class that the method is contained in. This allows the method to call variables that are contained within the class.

Data is the list of data that will be plotted with this method.

**Purpose:**
This method returns a string containing the maximum of the data that was passed through.

**Pre-Conditions:**
The data specified must not be null.
The data specified must have been initialized and have at least one piece of data in it.

**Post-Conditions:**
Returns a string containing the maximum of the data that was passed through.

**Signature:** sum(self, data = none)

**Method Name:** sum

**Type of return value:** Returns a string of containing the average of the data passed.

**Type of input parameters:** self, data is a list

**Type of output parameters:** none

**Description of input and output parameters:**
Self is the class that the method is contained in. This allows the method to call variables that are contained within the class.

Data is the list of data that will be plotted with this method.

**Purpose:**
This method returns a string containing the sum of the data that was passed through.

**Pre-Conditions:**
The data specified must not be null.
The data specified must have been initialized and have at least one piece of data in it.

**Post-Conditions:**
Returns a string containing the sum of the data that was passed through.

3.12 Class Description: R

**Concrete Class:** R

**Superclass:** Analysis Language

**Description:** This class is responsible for connecting to R, executing the specified function through R, and returning the results R produced.

**Private Responsibilities:**
Converts active data set to proper R commands
Submits the converted data (from #1 above) to R
3.12.1 Contract: Analysis Language 1

Since this class is a subclass of the Analysis Language class, it must abide by the contracts that the Analysis Language class holds. This class holds no other contracts, other than the ones specified by the superclass. See Section 3.11.1 for the details of the Analysis Language 1 Contract that the superclass holds.

3.13 Class Description: Reduction Technique

Abstract Class: Reduction Technique
Subclass: Average, Median, Mode, Kurtosis, Skew, Covariance, Histogram, Variable Correlation, Hypothesis Testing, Maximum, Minimum, Standard Deviation, Sum, Bar Graph, Scatter Plot, Line Graph, Density Chart, Box Chart, Pie Chart, Decision Making

1. Create instance(s) of appropriate Analysis Language by calling the appropriate function
   a. Passes active data set selection
2. Sends information to Execution Parser
   a. Passes results

Description: This class is responsible for enforcing a protocol for any reduction technique that might be used.

Private Responsibilities:
Knows the Analysis Language
Receives the active data set from Execution Parser \( EP (5) \)
Receives the active data set from Analysis Language \( AL (1) \)
Parses the parameters and arranges them into the order needed by R

3.13.1 Contract: Reduction Technique 1

Contract Identifier: RT1
Contract Name: Reduction Technique 1

The Reduction Technique class will follow the RT1 contract by sending the necessary data in the appropriate order to the reduction technique. The following is the getDatareduction method protocol.

Signature: getDatareduction(self)
Method Name: getDatareduction
Type of return value: Returns a string of the file path.
Type of input parameters: self
Type of output parameters: none

Description of input and output parameters:
Self is the class that the method is contained in. This allows the method to call variables that are contained within the class.

Purpose:
This method returns an image file for the reduction technique and data that is specified in the initialization of the class. This is an abstract method that is extended to each reduction technique to able to access the appropriate technique with one signature.

Pre-Conditions:
The data specified in the initialization must be reducible by the reduction technique that is specified in the initialization.
The data specified must not be null.
The data specified must have been initialized and have at least one piece of data in it.

Post-Conditions:
Does not delete, append or modify in any way, type or form the data used to produce the specific reduction technique.
Returns an image file for the reduction technique and data that is specified.

3.13.2 Contract: Reduction Technique 2

Contract Identifier: RT2
Contract Name: Reduction Technique 2
The Reduction Technique class will follow RT2 contract by providing a method that return the results of applying the specified function to the specified data utilizing R. For the getDataReduction method protocol see section 3.13.1.

3.14 Class Description: Average

Concrete Class: Average
Superclass: Reduction Technique
Description: This class is responsible passing the data to the average method in the R class and returning the results.
Private Responsibilities:
Responsible for calling appropriate function of the Analysis Language to perform an average calculation
Parses the parameters and arranges them into the order needed by R

3.14.1 Contract: Reduction Technique 1
Since this class is a subclass of the Reduction Technique class, it must abide by the contracts that the Reduction Technique class holds. This class holds no other contracts, other than the ones specified by the superclass. See Section 3.13.1 for the details of the Reduction Technique 1 Contract that its superclass holds.

3.14.2 Contract: Reduction Technique 2
Since this class is a subclass of the Reduction Technique class, it must abide by the contracts that the Reduction Technique class holds. This class holds no other contracts, other than the ones specified by the superclass. See Section 3.13.2 for the details of the Reduction Technique 2 Contract that its superclass holds.

3.15 Class Description: Median

Concrete Class: Median
Superclass: Reduction Technique
Description: This class is responsible passing the data to the median method in the R class and returning the results.
Private Responsibilities:
Responsible for calling appropriate function of the Analysis Language to perform a median calculation
Parses the parameters and arranges them into the order needed by R

3.15.1 Contract: Reduction Technique 1
Since this class is a subclass of the Reduction Technique class, it must abide by the contracts that the Reduction Technique class holds. This class holds no other contracts, other than the ones specified by the superclass. See Section 3.13.1 for the details of the Reduction Technique 1 Contract that its superclass holds.

3.15.2 Contract: Reduction Technique 2
Since this class is a subclass of the Reduction Technique class, it must abide by the contracts that the Reduction Technique class holds. This class holds no other contracts, other than the ones specified by the superclass. See Section 3.13.2 for the details of the Reduction Technique 2 Contract that its superclass holds.

3.16 Class Description: Mode

Concrete Class: Mode
Superclass: Reduction Technique
Description: This class is responsible passing the data to the modemethod in the R class and returning the results.
Private Responsibilities:
Responsible for calling appropriate function of the Analysis Language to perform a mode calculation
Parses the parameters and arranges them into the order needed by R
3.16.1 Contract: Reduction Technique 1
Since this class is a subclass of the Reduction Technique class, it must abide by the contracts that the Reduction Technique class holds. This class holds no other contracts, other than the ones specified by the superclass. See Section 3.13.1 for the details of the Reduction Technique 1 Contract that its superclass holds.

3.16.2 Contract: Reduction Technique 2
Since this class is a subclass of the Reduction Technique class, it must abide by the contracts that the Reduction Technique class holds. This class holds no other contracts, other than the ones specified by the superclass. See Section 3.13.2 for the details of the Reduction Technique 2 Contract that its superclass holds.

3.17 Class Description: Kurtosis
Concrete Class: Kurtosis
Superclass: Reduction Technique
Description: This class is responsible passing the data to the kurtosis method in the R class and returning the results.
Private Responsibilities:
Responsible for calling appropriate function of the Analysis Language to perform a kurtosis calculation
Parses the parameters and arranges them into the order needed by R

3.17.1 Contract: Reduction Technique 1
Since this class is a subclass of the Reduction Technique class, it must abide by the contracts that the Reduction Technique class holds. This class holds no other contracts, other than the ones specified by the superclass. See Section 3.13.1 for the details of the Reduction Technique 1 Contract that its superclass holds.

3.17.2 Contract: Reduction Technique 2
Since this class is a subclass of the Reduction Technique class, it must abide by the contracts that the Reduction Technique class holds. This class holds no other contracts, other than the ones specified by the superclass. See Section 3.13.2 for the details of the Reduction Technique 2 Contract that its superclass holds.

3.18 Class Description: Skew
Concrete Class: Skew
Superclass: Reduction Technique
Description: This class is responsible passing the data to the skew method in the R class and returning the results.
Private Responsibilities:
Responsible for calling appropriate function of the Analysis Language to perform a skew calculation
Parses the parameters and arranges them into the order needed by R

3.18.1 Contract: Reduction Technique 1
Since this class is a subclass of the Reduction Technique class, it must abide by the contracts that the Reduction Technique class holds. This class holds no other contracts, other than the ones specified by the superclass. See Section 3.13.1 for the details of the Reduction Technique 1 Contract that its superclass holds.

3.18.2 Contract: Reduction Technique 2
Since this class is a subclass of the Reduction Technique class, it must abide by the contracts that the Reduction Technique class holds. This class holds no other contracts, other than the ones specified by the superclass. See Section 3.13.2 for the details of the Reduction Technique 2 Contract that its superclass holds.

3.19 Class Description: Covariance
Concrete Class: Covariance
Superclass: Reduction Technique
Description: This class is responsible passing the data to the covariance method in the R class and returning the results.

Private Responsibilities:
Responsible for calling appropriate function of the Analysis Language to perform a covariance calculation
Parses the parameters and arranges them into the order needed by R

3.19.1 Contract: Reduction Technique 1
Since this class is a subclass of the Reduction Technique class, it must abide by the contracts that the Reduction Technique class holds. This class holds no other contracts, other than the ones specified by the superclass. See Section 3.13.1 for the details of the Reduction Technique 1 Contract that its superclass holds.

3.19.2 Contract: Reduction Technique 2
Since this class is a subclass of the Reduction Technique class, it must abide by the contracts that the Reduction Technique class holds. This class holds no other contracts, other than the ones specified by the superclass. See Section 3.13.2 for the details of the Reduction Technique 2 Contract that its superclass holds.

3.20 Class Description: Histogram Plot
Concrete Class: Histogram Plot
Superclass: Reduction Technique
Description: This class is responsible for parsing the parameters and arranging them in the order that R is expecting them in, calling the histogram plot method in the R class, and returning the results.

Private Responsibilities:
Responsible for calling appropriate function of the Analysis Language to produce a histogram plot
Parses the parameters and arranges them into the order needed by R

3.20.1 Contract: Reduction Technique 1
Since this class is a subclass of the Reduction Technique class, it must abide by the contracts that the Reduction Technique class holds. This class holds no other contracts, other than the ones specified by the superclass. See Section 3.13.1 for the details of the Reduction Technique 1 Contract that its superclass holds.

3.20.2 Contract: Reduction Technique 2
Since this class is a subclass of the Reduction Technique class, it must abide by the contracts that the Reduction Technique class holds. This class holds no other contracts, other than the ones specified by the superclass. See Section 3.13.2 for the details of the Reduction Technique 2 Contract that its superclass holds.

3.21 Class Description: Variable Correlation
Concrete Class: Variable Correlation
Superclass: Reduction Technique
Description: This class is responsible for parsing the parameters and arranging them in the order that R is expecting them in, calling the variable correlation method in the R class, and returning the results.

Private Responsibilities:
Responsible for calling appropriate function of the Analysis Language to perform a variable correlation calculation
Parses the parameters and arranges them into the order needed by R

3.21.1 Contract: Reduction Technique 1
Since this class is a subclass of the Reduction Technique class, it must abide by the contracts that the Reduction Technique class holds. This class holds no other contracts, other than the ones specified by the superclass. See Section 3.13.1 for the details of the Reduction Technique 1 Contract that its superclass holds.
3.21.2 Contract: Reduction Technique 2
Since this class is a subclass of the Reduction Technique class, it must abide by the contracts that the Reduction Technique class holds. This class holds no other contracts, other than the ones specified by the superclass. See Section 3.13.2 for the details of the Reduction Technique 2 Contract that its superclass holds.

3.22 Class Description: Hypothesis Testing
Concrete Class: Hypothesis Testing
Superclass: Reduction Technique
Description: This class is responsible for parsing the parameters and arranging them in the order that R is expecting them in, calling the hypothesis testing method in the R class, and returning the results.
Private Responsibilities:
- Responsible for calling appropriate function of the Analysis Language to perform a hypothesis testing calculation
- Parses the parameters and arranges them into the order needed by R

3.22.1 Contract: Reduction Technique 1
Since this class is a subclass of the Reduction Technique class, it must abide by the contracts that the Reduction Technique class holds. This class holds no other contracts, other than the ones specified by the superclass. See Section 3.13.1 for the details of the Reduction Technique 1 Contract that its superclass holds.

3.22.2 Contract: Reduction Technique 2
Since this class is a subclass of the Reduction Technique class, it must abide by the contracts that the Reduction Technique class holds. This class holds no other contracts, other than the ones specified by the superclass. See Section 3.13.2 for the details of the Reduction Technique 2 Contract that its superclass holds.

3.23 Class Description: Maximum
Concrete Class: Maximum
Superclass: Reduction Technique
Description: This class is responsible passing the data to the maximum method in the R class and returning the results.
Private Responsibilities:
- Responsible for calling appropriate function of the Analysis Language to perform a maximum calculation
- Parses the parameters and arranges them into the order needed by R

3.23.1 Contract: Reduction Technique 1
Since this class is a subclass of the Reduction Technique class, it must abide by the contracts that the Reduction Technique class holds. This class holds no other contracts, other than the ones specified by the superclass. See Section 3.13.1 for the details of the Reduction Technique 1 Contract that its superclass holds.

3.23.2 Contract: Reduction Technique 2
Since this class is a subclass of the Reduction Technique class, it must abide by the contracts that the Reduction Technique class holds. This class holds no other contracts, other than the ones specified by the superclass. See Section 3.13.2 for the details of the Reduction Technique 2 Contract that its superclass holds.

3.24 Class Description: Minimum
Concrete Class: Minimum
Superclass: Reduction Technique
Description: This class is responsible passing the data to the minimum method in the R class and returning the results.
Private Responsibilities:
- Responsible for calling appropriate function of the Analysis Language to perform a minimum calculation
Parses the parameters and arranges them into the order needed by R

### 3.24.1 Contract: Reduction Technique 1
Since this class is a subclass of the Reduction Technique class, it must abide by the contracts that the Reduction Technique class holds. This class holds no other contracts, other than the ones specified by the superclass. See Section 3.13.1 for the details of the Reduction Technique 1 Contract that its superclass holds.

### 3.24.2 Contract: Reduction Technique 2
Since this class is a subclass of the Reduction Technique class, it must abide by the contracts that the Reduction Technique class holds. This class holds no other contracts, other than the ones specified by the superclass. See Section 3.13.2 for the details of the Reduction Technique 2 Contract that its superclass holds.

### 3.25 Class Description: Standard Deviation
- **Concrete Class:** Standard Deviation
- **Superclass:** Reduction Technique
- **Description:** This class is responsible for passing the data to the standard deviation method in the R class and returning the results.
- **Private Responsibilities:**
  - Responsible for calling appropriate function of the Analysis Language to perform a standard deviation calculation
  - Parses the parameters and arranges them into the order needed by R

#### 3.25.1 Contract: Reduction Technique 1
Since this class is a subclass of the Reduction Technique class, it must abide by the contracts that the Reduction Technique class holds. This class holds no other contracts, other than the ones specified by the superclass. See Section 3.13.1 for the details of the Reduction Technique 1 Contract that its superclass holds.

#### 3.25.2 Contract: Reduction Technique 2
Since this class is a subclass of the Reduction Technique class, it must abide by the contracts that the Reduction Technique class holds. This class holds no other contracts, other than the ones specified by the superclass. See Section 3.13.2 for the details of the Reduction Technique 2 Contract that its superclass holds.

### 3.26 Class Description: Sum
- **Concrete Class:** Sum
- **Superclass:** Reduction Technique
- **Description:** This class is responsible for passing the data to the sum method in the R class and returning the results.
- **Private Responsibilities:**
  - Responsible for calling appropriate function of the Analysis Language to perform a sums calculation
  - Parses the parameters and arranges them into the order needed by R

#### 3.26.1 Contract: Reduction Technique 1
Since this class is a subclass of the Reduction Technique class, it must abide by the contracts that the Reduction Technique class holds. This class holds no other contracts, other than the ones specified by the superclass. See Section 3.13.1 for the details of the Reduction Technique 1 Contract that its superclass holds.

#### 3.26.2 Contract: Reduction Technique 2
Since this class is a subclass of the Reduction Technique class, it must abide by the contracts that the Reduction Technique class holds. This class holds no other contracts, other than the ones specified by the superclass. See Section 3.13.2 for the details of the Reduction Technique 2 Contract that its superclass holds.
3.27 Class Description: Bar Graph
Concrete Class: Bar Graph
Superclass: Reduction Technique
Description: This class is responsible for parsing the parameters and arranging them in the order that R is expecting them in, calling the bar graph method in the R class, and returning the results.
Private Responsibilities:
Responsible for calling appropriate function of the Analysis Language to produce a bar graph
Parses the parameters and arranges them into the order needed by R

3.27.1 Contract: Reduction Technique 1
Since this class is a subclass of the Reduction Technique class, it must abide by the contracts that the Reduction Technique class holds. This class holds no other contracts, other than the ones specified by the superclass. See Section 3.13.1 for the details of the Reduction Technique 1 Contract that its superclass holds.

3.27.2 Contract: Reduction Technique 2
Since this class is a subclass of the Reduction Technique class, it must abide by the contracts that the Reduction Technique class holds. This class holds no other contracts, other than the ones specified by the superclass. See Section 3.13.2 for the details of the Reduction Technique 2 Contract that its superclass holds.

3.28 Class Description: Scatter Plot
Concrete Class: Scatter Plot
Superclass: Reduction Technique
Description: This class is responsible for parsing the parameters and arranging them in the order that R is expecting them in, calling the scatter plot method in the R class, and returning the results.
Private Responsibilities:
Responsible for calling appropriate function of the Analysis Language to produce a scatter plot
Parses the parameters and arranges them into the order needed by R

3.28.1 Contract: Reduction Technique 1
Since this class is a subclass of the Reduction Technique class, it must abide by the contracts that the Reduction Technique class holds. This class holds no other contracts, other than the ones specified by the superclass. See Section 3.13.1 for the details of the Reduction Technique 1 Contract that its superclass holds.

3.28.2 Contract: Reduction Technique 2
Since this class is a subclass of the Reduction Technique class, it must abide by the contracts that the Reduction Technique class holds. This class holds no other contracts, other than the ones specified by the superclass. See Section 3.13.2 for the details of the Reduction Technique 2 Contract that its superclass holds.

3.29 Class Description: Line Graph
Concrete Class: Line Graph
Superclass: Reduction Technique
Description: This class is responsible for parsing the parameters and arranging them in the order that R is expecting them in, calling the line graph method in the R class, and returning the results.
Private Responsibilities:
Responsible for calling appropriate function of the Analysis Language to produce a line graph
Parses the parameters and arranges them into the order needed by R
3.29.1 Contract: Reduction Technique 1
Since this class is a subclass of the Reduction Technique class, it must abide by the contracts that the Reduction Technique class holds. This class holds no other contracts, other than the ones specified by the superclass. See Section 3.13.1 for the details of the Reduction Technique 1 Contract that its superclass holds.

3.29.2 Contract: Reduction Technique 2
Since this class is a subclass of the Reduction Technique class, it must abide by the contracts that the Reduction Technique class holds. This class holds no other contracts, other than the ones specified by the superclass. See Section 3.13.2 for the details of the Reduction Technique 2 Contract that its superclass holds.

3.30 Class Description: Density Chart
Concrete Class: Density Chart
Superclass: Reduction Technique
Description: This class is responsible for parsing the parameters and arranging them in the order that R is expecting them in, calling the density chart method in the R class, and returning the results.
Private Responsibilities:
Responsible for calling appropriate function of the Analysis Language to produce density charts
Parses the parameters and arranges them into the order needed by R

3.30.1 Contract: Reduction Technique 1
Since this class is a subclass of the Reduction Technique class, it must abide by the contracts that the Reduction Technique class holds. This class holds no other contracts, other than the ones specified by the superclass. See Section 3.13.1 for the details of the Reduction Technique 1 Contract that its superclass holds.

3.30.2 Contract: Reduction Technique 2
Since this class is a subclass of the Reduction Technique class, it must abide by the contracts that the Reduction Technique class holds. This class holds no other contracts, other than the ones specified by the superclass. See Section 3.13.2 for the details of the Reduction Technique 2 Contract that its superclass holds.

3.31 Class Description: Box Plot
Concrete Class: Box Plot
Superclass: Reduction Technique
Description: This class is responsible for parsing the parameters and arranging them in the order that R is expecting them in, calling the box plot method in the R class, and returning the results.
Private Responsibilities:
Responsible for calling appropriate function of the Analysis Language to produce box plot
Parses the parameters and arranges them into the order needed by R

3.31.1 Contract: Reduction Technique 1
Since this class is a subclass of the Reduction Technique class, it must abide by the contracts that the Reduction Technique class holds. This class holds no other contracts, other than the ones specified by the superclass. See Section 3.13.1 for the details of the Reduction Technique 1 Contract that its superclass holds.

3.31.2 Contract: Reduction Technique 2
Since this class is a subclass of the Reduction Technique class, it must abide by the contracts that the Reduction Technique class holds. This class holds no other contracts, other than the ones specified by the superclass. See Section 3.13.2 for the details of the Reduction Technique 2 Contract that its superclass holds.

3.32 Class Description: Pie Chart
Concrete Class: Pie Chart
Superclass: Reduction Technique

Description: This class is responsible for parsing the parameters and arranging them in the order that R is expecting them in, calling the pie chart method in the R class, and returning the results.

Private Responsibilities:
Responsible for calling appropriate function of the Analysis Language to produce a pie chart
Parses the parameters and arranges them into the order needed by R

3.32.1 Contract: Reduction Technique 1
Since this class is a subclass of the Reduction Technique class, it must abide by the contracts that the Reduction Technique class holds. This class holds no other contracts, other than the ones specified by the superclass. See Section 3.13.1 for the details of the Reduction Technique 1 Contract that its superclass holds.

3.32.2 Contract: Reduction Technique 2
Since this class is a subclass of the Reduction Technique class, it must abide by the contracts that the Reduction Technique class holds. This class holds no other contracts, other than the ones specified by the superclass. See Section 3.13.2 for the details of the Reduction Technique 2 Contract that its superclass holds.

3.33 Class Description: Decision Making
Concrete Class: Decision Making
Superclass: Reduction Technique
Description: This class is responsible for parsing the parameters and arranging them in the order that R is expecting them in, calling the decision making method in the R class, and returning the results.

Private Responsibilities:
Responsible for calling appropriate function of the Analysis Language to perform a decision making technique
Parses the parameters and arranges them into the order needed by R

3.33.1 Contract: Reduction Technique 1
Since this class is a subclass of the Reduction Technique class, it must abide by the contracts that the Reduction Technique class holds. This class holds no other contracts, other than the ones specified by the superclass. See Section 3.13.1 for the details of the Reduction Technique 1 Contract that its superclass holds.

3.33.2 Contract: Reduction Technique 2
Since this class is a subclass of the Reduction Technique class, it must abide by the contracts that the Reduction Technique class holds. This class holds no other contracts, other than the ones specified by the superclass. See Section 3.13.2 for the details of the Reduction Technique 2 Contract that its superclass holds.

3.34 Class Description: Classification
Concrete Class: Classification
Superclass: Reduction Technique
Description: This class is responsible for parsing the parameters and arranging them in the order that R is expecting them in, calling the classification method in the R class, and returning the results.

Private Responsibilities:
Responsible for calling appropriate function of the Analysis Language to perform a classification technique
Parses the parameters and arranges them into the order needed by R

3.34.1 Contract: Reduction Technique 1
Since this class is a subclass of the Reduction Technique class, it must abide by the contracts that the Reduction Technique class holds. This class holds no other contracts, other than the ones specified by the superclass. See Section 3.13.1 for the details of the Reduction Technique 1 Contract that its superclass holds.
3.34.2 Contract: Reduction Technique 2
Since this class is a subclass of the Reduction Technique class, it must abide by the contracts that the Reduction Technique class holds. This class holds no other contracts, other than the ones specified by the superclass. See Section 3.13.2 for the details of the Reduction Technique 2 Contract that its superclass holds.

3.35 Class Description: Strip Plot
Concrete Class: Strip Plot
Superclass: Reduction Technique
Description: This class is responsible for parsing the parameters and arranging them in the order that R is expecting them in, calling the strip plot method in the R class, and returning the results.
Private Responsibilities:
Responsible for calling appropriate function of the Analysis Language to produce a strip plot technique
Parses the parameters and arranges them into the order needed by R

3.35.1 Contract: Reduction Technique 1
Since this class is a subclass of the Reduction Technique class, it must abide by the contracts that the Reduction Technique class holds. This class holds no other contracts, other than the ones specified by the superclass. See Section 3.13.1 for the details of the Reduction Technique 1 Contract that its superclass holds.

3.35.2 Contract: Reduction Technique 2
Since this class is a subclass of the Reduction Technique class, it must abide by the contracts that the Reduction Technique class holds. This class holds no other contracts, other than the ones specified by the superclass. See Section 3.13.2 for the details of the Reduction Technique 2 Contract that its superclass holds.

3.36 Class Description: Regression
Concrete Class: Regression
Superclass: Reduction Technique
Description: This class is responsible for parsing the parameters and arranging them in the order that R is expecting them in, calling the regression method in the R class, and returning the results.
Private Responsibilities:
Responsible for calling appropriate function of the Analysis Language to produce a regression
Parses the parameters and arranges them into the order needed by R

3.36.1 Contract: Reduction Technique 1
Since this class is a subclass of the Reduction Technique class, it must abide by the contracts that the Reduction Technique class holds. This class holds no other contracts, other than the ones specified by the superclass. See Section 3.13.1 for the details of the Reduction Technique 1 Contract that its superclass holds.

3.36.2 Contract: Reduction Technique 2
Since this class is a subclass of the Reduction Technique class, it must abide by the contracts that the Reduction Technique class holds. This class holds no other contracts, other than the ones specified by the superclass. See Section 3.13.2 for the details of the Reduction Technique 2 Contract that its superclass holds.

Q.E.D.