|  |
| --- |
| **Software Design Specification** |

Matrix Service API

Create Date: 3/8/2012

Created By: Sarah Kreidler

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[**Software Design Specification** 1](#_Toc321484833)

[1 Introduction 4](#_Toc321484834)

[1.1 Purpose of this document 4](#_Toc321484835)

[1.2 Scope of the development project 4](#_Toc321484836)

[1.3 Definitions, Acronyms, and Abbreviations 4](#_Toc321484837)

[1.4 References 4](#_Toc321484838)

[1.5 Overview of the document 4](#_Toc321484839)

[2 System Architecture Description 5](#_Toc321484840)

[2.1 Overview of modules / components 5](#_Toc321484841)

[2.2 Structure and relationships 5](#_Toc321484842)

[3 Module and Component Descriptions 5](#_Toc321484843)

[3.1 Component overview 5](#_Toc321484844)

[3.2 External API 5](#_Toc321484845)

[3.2.1 Domain Objects 5](#_Toc321484846)

[3.3 The Resource Classes 6](#_Toc321484847)

[3.3.1 The Matrix Addition Resource 6](#_Toc321484848)

[3.3.2 The Matrix Subtraction Resource 6](#_Toc321484849)

[3.3.3 The Matrix Multiplication Resource 6](#_Toc321484850)

[3.3.4 The Matrix Element Wise Multiplication Resource 6](#_Toc321484851)

[3.3.5 The Matrix Horizontal Direct Multiply Resource 6](#_Toc321484852)

[3.3.6 The Matrix Scalar Multiply Resource 7](#_Toc321484853)

[3.3.7 The Matrix Kronecker Multiply Resource 7](#_Toc321484854)

[3.3.8 The Matrix Cholesky Decompose Resource 7](#_Toc321484855)

[3.3.9 The Matrix Inversion Resource 7](#_Toc321484856)

[3.3.10 The Matrix Rank Resource 8](#_Toc321484857)

[3.3.11 The Matrix Trace Resource 8](#_Toc321484858)

[3.3.12 The Matrix Is Positive Definite Resource 8](#_Toc321484859)

[3.3.13 The Matrix Vec Resource 8](#_Toc321484860)

[3.3.14 The Matrix Vech Resource 8](#_Toc321484861)

[3.3.15 The Contrast Get Between Interaction Contrast Resource 9](#_Toc321484862)

[3.3.16 The Contrast Get Within Interaction Contrast Resource 9](#_Toc321484863)

[3.3.17 The Contrast Get Between Main Effect Contrast Resource 9](#_Toc321484864)

[3.3.18 The Contrast Get Within Main Effect Contrast Resource 9](#_Toc321484865)

[3.3.19 The Contrast Get Between Grand Mean Contrast Resource 9](#_Toc321484866)

[3.3.20 The Contrast Get Within Grand Mean Contrast Resource 9](#_Toc321484867)

[3.3.21 The Contrast Get Orthogonal Polynomial Coefficients 9](#_Toc321484868)

[3.4 Exception Handling 11](#_Toc321484869)

[3.5 Unit Testing 11](#_Toc321484870)

[3.6 Recommended Client Software 11](#_Toc321484871)

[3.6.1 Example call to the Server Resources 12](#_Toc321484872)

[4 Reuse and Relationships to Other Products 12](#_Toc321484873)

[4.1 Reuse of existing code 12](#_Toc321484874)

[5 Design Trade-offs 12](#_Toc321484875)

[5.1 Why Restlet ConverterService based communication? 12](#_Toc321484876)

[5.2 Why use the NamedMatrix object instead of Apache RealMatrix? 13](#_Toc321484877)

# Introduction

## Purpose of this document

This document describes proposed changes to the Matrix Service API. The Matrix Service is a Java Web Service which provides matrix calculations such as matrix arithmetic, rank, and construction of orthogonal polynomials. The Matrix Service is accessed by the GLIMMPSE user interface.

The proposed changes include

* Replacing the existing XML API with Restlet ConverterService API1

## Scope of the development project

The development will be limited to the Matrix Service module.

## Definitions, Acronyms, and Abbreviations

**XML**2- Extensible Markup Language (XML) is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable.

**AJAX** – asynchronous HTTP request. In this context, AJAX requests are issued to update the study design information with the Study Design Service, or to perform a matrix operation.

**Restlet**1 – A lightweight framework for writing web services and managing client/server interactions.

## References

1. Anon. Restlet 2.0 - Documentation. Available at: http://www.restlet.org/documentation/2.0/. Accessed March 9, 2012.

2. Anon. Extensible Markup Language (XML) 1.0 (Fifth Edition). Available at: http://www.w3.org/TR/REC-xml/. Accessed February 10, 2012.

3. Noelios Technologies. Restlet. Version 1.1.6. 2009. Available at: http://www.restlet.org/. Accessed May 17, 2011.

4. Apache Software Foundation. Apache Tomcat. Version 6.0. Available at: http://tomcat.apache.org/. Accessed May 17, 2011.

5. Sakhadeo U, Kreidler S. Software Design Specification: GLIMMPSE Domain Objects. 2011.

6. Muller KE, Stewart PW. *Linear model theory: univariate, multivariate, and mixed models*. Hoboken, New Jersey: John Wiley and Sons; 2006.

7. Anon. First application. Available at: http://wiki.restlet.org/docs\_2.0/13-restlet/21-restlet/318-restlet/303-restlet.html. Accessed March 9, 2012.

## Overview of the document

We describe the integration of the Matrix Service with the overall GLIMMPSE architecture. We define the available function calls in the Matrix Service and address error handling.

# System Architecture Description

## Overview of modules / components

The Matrix Service is a stand-alone web service which provides linear algebra operations which are difficult to perform within the user interface. The separation of matrix calculations avoids JavaScript limitations in the browser-based interface of GLIMMPSE.

## Structure and relationships

The Matrix Service has not been formally released, so no other GLIMMPSE components currently access the service. GLIMMPSE version 2.0.0 will begin using the Matrix Service for validation of matrices in Matrix Mode, and for construction of complex contrasts in trend hypotheses.

We intend to branch Matrix Service 1.0.0 to preserve the XML interface. We will release the RPC-style interface as version 2.0.0.

# Module and Component Descriptions

## Component overview

The Matrix Service will be implemented using the Restlet3 library version 2.0.x. The service will provide linear algebra operations for matrices or lists of matrices. In addition, it will provide an API for generating polynomial contrasts.

The service will be deployed within an Apache Tomcat server4 using the standard warfile format.

## External API

### Domain Objects

The Matrix Service takes the following objects as inputs:

* NamedMatrix objects – primary matrix representation for linear algebra operations
* List of BetweenParticipantFactor objects – used for generating contrasts to test trend hypotheses on between participant effects
* List of RepeatedMeasuresNode objects – used for generating contrasts to test trend hypotheses on within participant effects

Returned objects include

* Lists of NamedMatrix objects
* OrthogonalPolynomialContrastCollection objects – collection of possible contrasts based on a list of factors
* Integer , Double, Boolean objects – for descriptive operations such as rank or trace

The objects are defined in detail in the Domain Layer design document5.

## The Resource Classes

The Resource Classes are the external interface for performing matrix operations. See Chapter 1 of Muller and Stewart6 for details on matrix operations and matrix dimension restrictions. The following are the Resource classes for the Matrix operations needed.

### The Matrix Addition Resource

The Matrix Addition Resource is the external interface for performing matrix addition operation. The following is the method to perform this operation.

Method: add(ArrayList<NamedMatrix> matrixList)

Returns: NamedMatrix

URI to call the Matrix Addition Server Resource class is

http://tomcat-server/matrix/addition

### The Matrix Subtraction Resource

The Matrix SubtractionResource is the external interface for performing matrix subtraction operation. The following is the method to perform this operation.

Method: subtract(ArrayList<NamedMatrix> matrixList)

Returns: NamedMatrix

URI to call the Matrix SubtractionServer Resource class is

http://tomcat-server/matrix/subtraction

### The Matrix Multiplication Resource

The Matrix Multiplication Resource is the external interface for performing matrix multiplication operation. The following is the method to perform this operation.

Method: multiply(ArrayList<NamedMatrix> matrixList)

Returns: NamedMatrix

URI to call the Matrix Multiplication Server Resource class is

http://tomcat-server/matrix/mult

### The Matrix Element Wise Multiplication Resource

The Matrix Element Wise MultiplicationResource is the external interface for performing matrix element wise multiplication operation. The following is the method to perform this operation.

Method: elementWiseMultiply (ArrayList<NamedMatrix> matrixList)

Returns: NamedMatrix

URI to call the Matrix Element Wise Multiplication Server Resource class is

http://tomcat-server/matrix/mult/elementWise

### The Matrix Horizontal Direct Multiply Resource

The Matrix Horizontal Direct MultiplyResource is the external interface for performing matrix horizontal direct multiplication operation. The following is the method to perform this operation.

Method: horizontalDirectMultiply(ArrayList<NamedMatrix> matrixList)

Returns: NamedMatrix

URI to call the Matrix Horizontal Direct Multiply Server Resource class is

http://tomcat-server/matrix/mult/horizontalDir

### The Matrix Scalar Multiply Resource

The Matrix Scalar MultiplyResource is the external interface for performing Scalar multiplication operation. The following is the method to perform this operation

Method: scalarMultiply(double scalar, NamedMatrix matrix)

Returns: NamedMatrix

URI to call the Matrix Scalar Multiply Server Resource class is

http://tomcat-server/matrix/mult/scalar

### The Matrix Kronecker Multiply Resource

The Matrix Kronecker MultiplyResource is the external interface for performing Kronecker multiplication operation. The following is the method to perform this operation.

Method: kroneckerMultiply(ArrayList<NamedMatrix> matrixList)

Returns: NamedMatrix

URI to call the Matrix Kronecker Multiply Server Resource class is

http://tomcat-server/matrix/mult/kronecker

### The Matrix Cholesky Decompose Resource

The Matrix Cholesky MultiplyResource is the external interface for performing Cholesky decomposition operation. The following is the method to perform this operation.

Method: choleskyDecompose(NamedMatrix matrix)

Returns: ArrayList<NamedMatrix>

URI to call the Matrix Cholesky Decompose Server Resource class is

http://tomcat-server/matrix/decomposition/cholesky

### The Matrix Inversion Resource

The Matrix Inversion Resource is the external interface for performing matrix inverse. The following is the method to perform this operation.

Method: invert(NamedMatrix matrix)

Returns: NamedMatrix

URI to call the Matrix Inversion Server Resource class is

http://tomcat-server/matrix/inverse

### The Matrix Rank Resource

The Matrix Rank Resource is the external interface for determining rank of a matrix. The following is the method to perform this operation.

Method: rank(NamedMatrix matrix)

Returns: Integer

URI to call the Matrix Rank Server Resource class is

http://tomcat-server/matrix/rank

### The Matrix Trace Resource

The Matrix Trace Resource is the external interface for determining trace of a matrix. The following is the method to perform this operation.

Method: trace(NamedMatrix matrix)

Returns: Double

URI to call the Matrix Trace Server Resource class is

http://tomcat-server/matrix/trace

### The Matrix Is Positive Definite Resource

The Matrix Is Positive Resource is the external interface for determining if a matrix is positive definite. The following is the method to perform this operation.

Method: isPositiveDefinite(NamedMatrix matrix)

Returns: Boolean

URI to call the Matrix Is Positive Server Resource class is

http://tomcat-server/matrix/positiveDefinite

### The Matrix Vec Resource

The Matrix Vec Resource is the external interface for determining Vec matrix of a matrix. The following is the method to perform this operation.

Method: vec(NamedMatrix matrix)

Returns: NamedMatrix

URI to call the Matrix Vec Server Resource class is

http://tomcat-server/matrix/vec

### The Matrix Vech Resource

The Matrix Vech Resource is the external interface for creating a column vector whose elements are the stacked columns of the lower triangular elements of the matrix. The following is the method to perform this operation.

Method: vech(NamedMatrix matrix)

Returns: NamedMatrix

URI to call the Matrix Vech Server Resource class is

http://tomcat-server/matrix/vech

### The Contrast Get Between Interaction Contrast Resource

The Contrast Get Between Interaction Contrast Resource is the external interface for getting a polynomial contrast between all the Between Participants and Between Participants test factor list.

Method: getBetweenInteractionContrast(

ArrayList<BetweenParticipantFactor> fullFactorList,

ArrayList<BetweenParticipantFactor> testFactorList)

Returns: NamedMatrix

URI to call the Contrast Get Between Interaction Contrast Server Resource class is

http://tomcat-server/contrast/interaction/between

### The Contrast Get Within Interaction Contrast Resource

The Contrast Get Within Interaction Contrast Resource is the external interface for getting a polynomial contrast between all the Repeated Measures Nodes and Repeated Measures Test Factor nodes.

Method: getWithinInteractionContrast(

ArrayList<RepeatedMeasuresNode> fullFactorList,

ArrayList<RepeatedMeasuresNode> testFactorList)

Returns: NamedMatrix

URI to call the Contrast Get Within Contrast Server Resource class is

http://tomcat-server/contrast/interaction/within

### The Contrast Get Between Main Effect Contrast Resource

The Contrast Get Between Main Effect Contrast Resource is the external interface for getting a polynomial contrast between all the Between Participants and Between Participants test factor list.

Method: getContrast(

ArrayList<BetweenParticipantFactor> fullFactorList,

ArrayList<BetweenParticipantFactor> testFactorList)

Returns: NamedMatrix

URI to call the Contrast Get Between MainEffect Contrast Server Resource class is

http://tomcat-server/contrast/interaction/between

### The Contrast Get Within Main Effect Contrast Resource

The Contrast Get Within Main Effect Contrast Resource is the external interface for getting a polynomial contrast between all the Repeated Measures Node and Repeated Measures nodes test factor list.

Method: getContrast(

ArrayList<RepeatedMeasuresNode> fullFactorList,

ArrayList< RepeatedMeasuresNode > testFactorList)

Returns: NamedMatrix

URI to call the Contrast Get Within Main Effect Contrast Server Resource class is

http://tomcat-server/contrast/interaction/within

### The Contrast Get Between Grand Mean Contrast Resource

The Contrast Get Between Grand Mean Contrast Resource is an external interface for getting a polynomial contrast of a specified list of Between Participant Factors.

Method: getBetweenGrandMeanContrast(

ArrayList<BetweenParticipantFactor> fullFactorList)

Returns: NamedMatrix

URI to call Contrast Get Between Grand Mean Contrast Server Resource class is

http://tomcat-server/contrast/grandmean/between

### The Contrast Get Within Grand Mean Contrast Resource

The Contrast Get Within Grand Mean Contrast Resource is an external interface for getting a polynomial contrast of a specified list of Repeated Measures Nodes.

Method: getWithinGrandMeanContrast(

ArrayList<RepeatedMeasuresNode> fullFactorList)

Returns: NamedMatrix

URI to call Contrast Get Within Grand Mean Contrast Server Resource class is

http://tomcat-server/contrast/grandmean/within

### The Contrast Get Orthogonal Polynomial Coefficients

The Contrast Get Orthogonal Polynomial Coefficients Resource is an external interface for computing orthogonal polynomial contrasts for the specified data values.

Method: getOrthogonalPolynomialCoefficients(double[] x, int maxDegree)

Returns: NamedMatrix

URI to call Contrast Get Orthogonal Polynomial Coefficients Server Resource class is

http://tomcat-server/contrast/coefficients

## Exception Handling

When an error occurs in the web service, the RPC function will throw an exception with associated error message and error code. The Restlet library recommends adding the following client code to catch the error:

## Unit Testing

Unit tests will be developed using the JUnit framework. The unit tests will test each function call in the Matrix Resource and Contrast Resource API.

## Recommended Client Software

The Restlet3 library provides a convenient client interface for RPC. The client library is available for applications written in Google Web Toolkit, Java, and Android.

### Example call to the Server Resources

The following calls will perform a Matrix Addition Server Resource class.

List<NamedMatrix> matricesToAdd …; // initialization not shown

// Initialize the resource proxy.

ClientResource cr = new ClientResource("http://tomcat-server/matrix/add");

MatrixResource resource = cr.wrap(MatrixResource.class);

// add the list of matrices

NamedMatrix sumMatrix = resource.add(matricesToAdd);

# Reuse and Relationships to Other Products

## Reuse of existing code

The Matrix Service utilizes the JavaStatistics library and Apache Commons Math for matrix operations. The HTTP communication layer is handled by the Restlet libraries.

# Design Trade-offs

## Why Restlet ConverterService based communication?

The original API for Matrix Service was written in XML. This was chosen due to familiarity and ease of use with the Restlet 1.1.6 libraries3. In the XML implementation both the client and server would have to package Java objects as XML prior to communication. This added a significant amount of code on both the GLIMMPSE user interface and Matrix Service. Although XML is more easily readable, it is very verbose. In addition, parsing the DOM tree is a potential source of coding errors.

The Restlet 2.0.x libraries1 now support multiple transmission formats include XML, JSON, and serialized Java objects. We decided to follow the model provided in the “First Application” tutorial7 in the Restlet documentation. This uses serialized Java objects as the underlying communication format. The advantages to this approach are

* Lighter weight transmission of Java objects
* No parsing code required
* External API resembles a Java library, rather than an HTTP request.
* Client code is already provided in the Restlet library
  + Currently supported for Java applications, Google Web Toolkit, and Android mobile devices
* Java is platform independent
* May provide greater numerical stability as opposed to converting small floating point values to strings as in the XML approach.

The primary disadvantage is that our code will be tied to Restlet. However, this product is a software industry standard and has extensive documentation. We believe the rapid development benefits outweigh the risks with this approach.

## Why use the NamedMatrix object instead of Apache RealMatrix?

The NamedMatrix object is a server and client independent matrix representation. Certain client applications (such as Google Web Toolkit) require that all serialized objects implement concrete classes. Therefore, we cannot transmit matrices using known serializable classes such as Apache’s RealMatrix interface. The NamedMatrix object is a simple representation of a matrix using simple types (lists, doubles, etc.) which should be available for most client applications. Unfortunately, this requires translation on the server side into RealMatrix objects.