|  |
| --- |
| **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](#_Toc318998418)

[1 Introduction 3](#_Toc318998419)

[1.1 Purpose of this document 3](#_Toc318998420)

[1.2 Scope of the development project 3](#_Toc318998421)

[1.3 Definitions, Acronyms, and Abbreviations 3](#_Toc318998422)

[1.4 References 3](#_Toc318998423)

[1.5 Overview of the document 4](#_Toc318998424)

[2 System Architecture Description 4](#_Toc318998425)

[2.1 Overview of modules / components 4](#_Toc318998426)

[2.2 Structure and relationships 4](#_Toc318998427)

[3 Module and Component Descriptions 4](#_Toc318998428)

[3.1 Component overview 4](#_Toc318998429)

[3.2 External API 4](#_Toc318998430)

[3.2.1 Domain Objects 4](#_Toc318998431)

[3.2.2 The Matrix Resource 5](#_Toc318998432)

[3.2.3 The Contrast Resource 6](#_Toc318998433)

[3.3 Exception Handling 6](#_Toc318998434)

[3.4 Unit Testing 6](#_Toc318998435)

[3.5 Recommended Client Software 6](#_Toc318998436)

[3.5.1 Example call to the Matrix Server Resource 6](#_Toc318998437)

[4 Reuse and Relationships to Other Products 6](#_Toc318998438)

[4.1 Reuse of existing code 7](#_Toc318998439)

[5 Design Trade-offs 7](#_Toc318998440)

[5.1 Why Restlet ConverterService based communication? 7](#_Toc318998441)

# 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 Matrix Resource

The Matrix Resource is 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 function calls are supported.

|  |  |
| --- | --- |
| Method Summary | |
| NamedMatrix | add(ArrayList<NamedMatrix> matrixList)  Return the sum of the matrices in the input list. |
| NamedMatrix | subtract(ArrayList<NamedMatrix> matrixList)  Return the difference of the matrices in the input list. |
| NamedMatrix | multiply(ArrayList<NamedMatrix> matrixList)  Return the product of the matrices in the input list. |
| NamedMatrix | elementWiseMultiply(ArrayList<NamedMatrix> matrixList)  Return the element wise product of the matrices in the input list. |
| NamedMatrix | horizontalDirectMultiply(ArrayList<NamedMatrix> matrixList)  Return the horizontal direct product of the matrices in the input list. |
| NamedMatrix | scalarMultiply(double scalar, NamedMatrix matrix)  Return the product of the matrix with the specified scalar value. |
| NamedMatrix | kroneckerMultiply(ArrayList<NamedMatrix> matrixList)  Return the kronecker product of the matrices in the input list. |
| ArrayList<NamedMatrix> | choleskyDecompose(NamedMatrix matrix)  Return the Cholesky decomposition of the matrix. The returned list of matrices will include both components of the decomposition, named “L” and “Ltranspose” respectively. |
| NamedMatrix | invert(NamedMatrix matrix)  Return the inverse of the matrix if possible. Throws an exception for singular matrices. |
| Integer | rank(NamedMatrix matrix)  Return the rank of the matrix. |
| Double | trace(NamedMatrix matrix)  Return the trace of the matrix. |
| Boolean | isPositiveDefinite(NamedMatrix matrix)  Return true if the matrix is positive definite. |
| NamedMatrix | vec(NamedMatrix matrix)  Return the vec of the matrix. |
| NamedMatrix | vech(NamedMatrix matrix)  Return the vech of the matrix. |

### The Contrast Resource

The contrast resource allows users to generate complex polynomial contrasts for between or within participant factors. The resource also allows users to request orthogonal polynomial coefficients. The following functions are supported.

|  |  |
| --- | --- |
| Method Summary | |
| NamedMatrix | getBetweenInteractionContrast(  ArrayList<BetweenParticipantFactor> fullFactorList,  ArrayList<BetweenParticipantFactor> testFactorList)  Return a polynomial contrast to test the interaction of the specified factors. Note that both the full set of factors and subset of factors to be tested must be specified to ensure a contrast matrix of appropriate dimension. |
| NamedMatrix | getWithinInteractionContrast(  ArrayList<RepeatedMeasuresNode> fullFactorList,  ArrayList<RepeatedMeasuresNode> testFactorList)  Return a polynomial contrast to test the interaction of the specified factors. Note that both the full set of factors and subset of factors to be tested must be specified to ensure a contrast matrix of appropriate dimension. |
| NamedMatrix | getMainEffectContrast(  ArrayList<BetweenParticipantFactor> fullFactorList,  BetweenParticipantFactor testFactor)  Return a polynomial contrast to test the main effect of the specified factor. Note that both the full set of factors and subset of factors to be tested must be specified to ensure a contrast matrix of appropriate dimension. |
| NamedMatrix | getMainEffectContrast(  ArrayList<RepeatedMeasuresNode> fullFactorList,  RepeatedMeasuresNode testFactor)  Return a polynomial contrast to test the main effect of the specified factor. Note that both the full set of factors and subset of factors to be tested must be specified to ensure a contrast matrix of appropriate dimension. |
| NamedMatrix | getBetweenGrandMeanContrast(  ArrayList<BetweenParticipantFactor> fullFactorList)  Return a polynomial contrast to test the grand mean for the specified list of factors. |
| NamedMatrix | getWithinGrandMeanContrast(  ArrayList<RepeatedMeasuresNode> fullFactorList)  Return a polynomial contrast to test the grand mean for the specified list of factors. |
| NamedMatrix | getOrthogonalPolynomialCoefficients(double[] x, int maxDegree)  Computes orthogonal polynomial contrasts for the specified data values. |

## 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 Matrix Server Resource

The following calls will perform a matrix addition of a list of NamedMatrix objects.

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

// Initialize the resource proxy.

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

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.