TR-2005-[ID]

Data Retrieval Tools

Software Design Specification

Jun Peng and Kincho H. Law

Stanford University

Last Modified: 2005-03-25 Draft: 1.0

Acknowledgment: This work was supported by the George E. Brown, Jr. Network for Earthquake Engineering Simulation (NEES) Program of the National Science Foundation under Award Number CMS-0402490. Visit http://it.nees.org/ for more information.

Table of Contents

List of Figures

1 Introduction

This document describes a set of software tools to demonstrate the retrieval of experimental data archived in the NEESgrid framework. One of the key services that NEESgrid needs to support is the management of data and metadata for earthquake engineering simulations. To facilitate data management and access, reference data models (data schema) have been proposed and developed (Peng and Law 2004a; 2004b). The data models include the relationships among the data archived in the NEESgrid framework. The data retrieval tools take advantage of the relationships defined in the data models to retrieve related data.

A data model is in essence a representation of the data and their interrelationship and provides a conceptual or implementation view of the data and data management. In general, the structure of a data model can be described in many different forms, and is most commonly expressed using relational or object-oriented models. Relational modeling approach organizes data into entities and defines the relationships among the entities. The data in this model typically are stored in relational databases, such as Oracle, DB2 or MySQL. In an object-oriented data model, information is modeled as objects, which are organized as any sorts of (real or abstract) entities. Different data formats can be used to represent data objects and the model. For hierarchical structured object models, the semi-structured data is often expressed using XML (Extensible Markup Language) or the related frameworks, such as RDF (Resource Description Framework) or OWL (Web Ontology Language).

Current NEESgrid reference data models have been defined in both object-oriented and relational models. The reference data models can be utilized to facilitate the design of a data repository for storing the data and related metadata (data about data). The data can be stored using a database system or a file system. Most of the common file formats (such as Word, PPT, Excel, PDF, TXT, CAD drawing, JPEG, MPEG, AVI, etc.) are supported. For storing the metadata, XML-based system is employed for the object-oriented data model, while a relational database system is employed for relational data model.

Once the data and the metadata are saved in a repository, software tools are needed to retrieve them. A set of demonstrational tools are developed to support browsing and retrieving data that are saved according to the reference data models. These software tools serve to demonstrate the potential usage of a data repository and the process in accessing the data. Since a data model defines the structure and the relationships among the data, related data in the repository can be grouped and retrieved. These related data can be reorganized to support further engineering needs such as to facilitate organizing the data according to the structure of an engineering report.

1.1 User Problem Statement

One purpose of the data repository is to support data-driven scientific applications in the NEESgrid. Currently, NEES users do not have a simple way to browse and retrieve experimental data and metadata. Typically, a user is interested in accessing the data and the metadata related to a particular project. The relationships among the data and the metadata need to be explored so that data pertaining to a project can be organized and presented to the user in a meaningful and easy-to-understand manner.

1.2 Design Overview

As of this writing, the design of the data repository for NEES has yet to be finalized (Warnock 2005). The data retrieval tools need to be sufficiently flexible to cope with the changes for the data repository. Both an object data model and a relational data model are developed to demonstrate the tools for different data repositories. The software tools are thus developed to browse and retrieve the data that are organized in either object-oriented models or relational models.

There are two typical methods for retrieving the data from a data repository, browsing and querying. The data retrieval tools support data browsing by displaying the fields of an object. The fields of an object include both the content and the links to other objects. The content of an object can be displayed directly and the links can be followed to access other objects. However, query functions (analogous to SQL query for relational databases) have not been implemented. Instead, users can use pre-defined access functions to retrieve related information in the repository. The data access functions are built following typical user practice, and a demonstration application is built based on an engineering report structure provided by Prof. Andrei Reinhorn of SUNY-Buffalo.

The current design of the software tools also allows data retrieval from different data repositories. As shown in Figure 1, a single, unified interface is designed as an entry point to access data saved in two object file systems and a relational database system. The complexity of the underlying data structure and the distributed data storage are transparent to the users. That is, the data retrieval tools serve as a data integration service to present the users with a single access point.

1.3 System Requirements

As noted, the current implementation supports both semi-structured object data model and relational data model. OWL (Web Ontology Language) files are used to store the data for the object-oriented model and MySQL database is employed to store the data for the relational model. Several external tools and libraries are employed for the development and/or the execution of these data retrieval tools. They include:

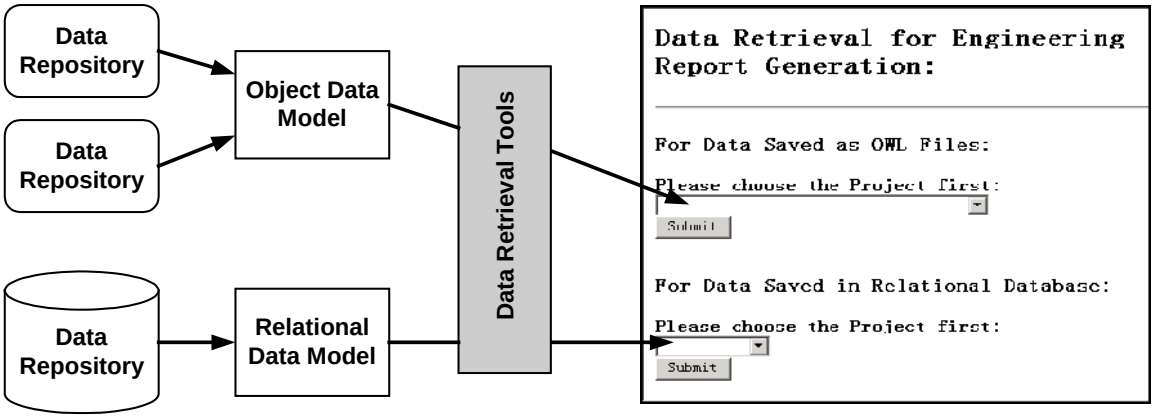

Figure 1: Design of Data Retrieval Tools

- Jena API to read, write and manipulate OWL files.
- MySQL database to store data in relational form.
- MySQL Connector J/3.0 (JDBC implementation) to connect Java programs with the MySQL database.
- Java Servlet for retrieving data from the data repository and dynamically generating web pages.

1.4 Integration Issues

The data retrieval tools are developed according to the employed central data repository. Depending on a data repository using OWL-based file system or MySQL relational database system and its location, the data retrieval tools need to be configured accordingly. The interface to the data retrieval tools is standard web browsers and no specific setup for web browsers is needed. The data retrieval tools are not dependent on any other NEESgrid software systems.

1.5 Terms and Definitions

OWL – Web Ontology Language. OWL can be used to explicitly represent the meaning of terms and the relationships between those terms. OWL is represented in XML format. http://www.w3.org/2001/sw/WebOnt/.

Jena – A Semantic Web Framework for Java. Jena provides a programmatic environment for OWL. By using Jena, OWL documents can be parsed and interpreted. http://jena.sourceforge.net/.

JDBC – Java Database Connectivity. JDBC is part of the Java Development Kit (JDK) that defines an API (application program interface) to support standard SQL access to databases from Java programs. http://java.sun.com/products/jdbc/.

2 System Architecture

Figure 2 shows a multi-layered design of the data retrieval system. There are several benefits of the layered design. First, each layer is responsible for specific set of functions and the components within the layer are self-contained. Therefore, each component can be designed and developed separately. Second, the components or modules in each layer depend only on the functions provided by the layer below it. This modular approach helps hiding the system complexity during the development process and facilitates debugging the system. Most importantly, the layered design allows replacing a module with minimizing effects on other modules. For example, if a new OWL parsing tool becomes available, we can replace Jena (shown in Figure 2) with the new tool and only the component OWLAccess.java needs to be modified.

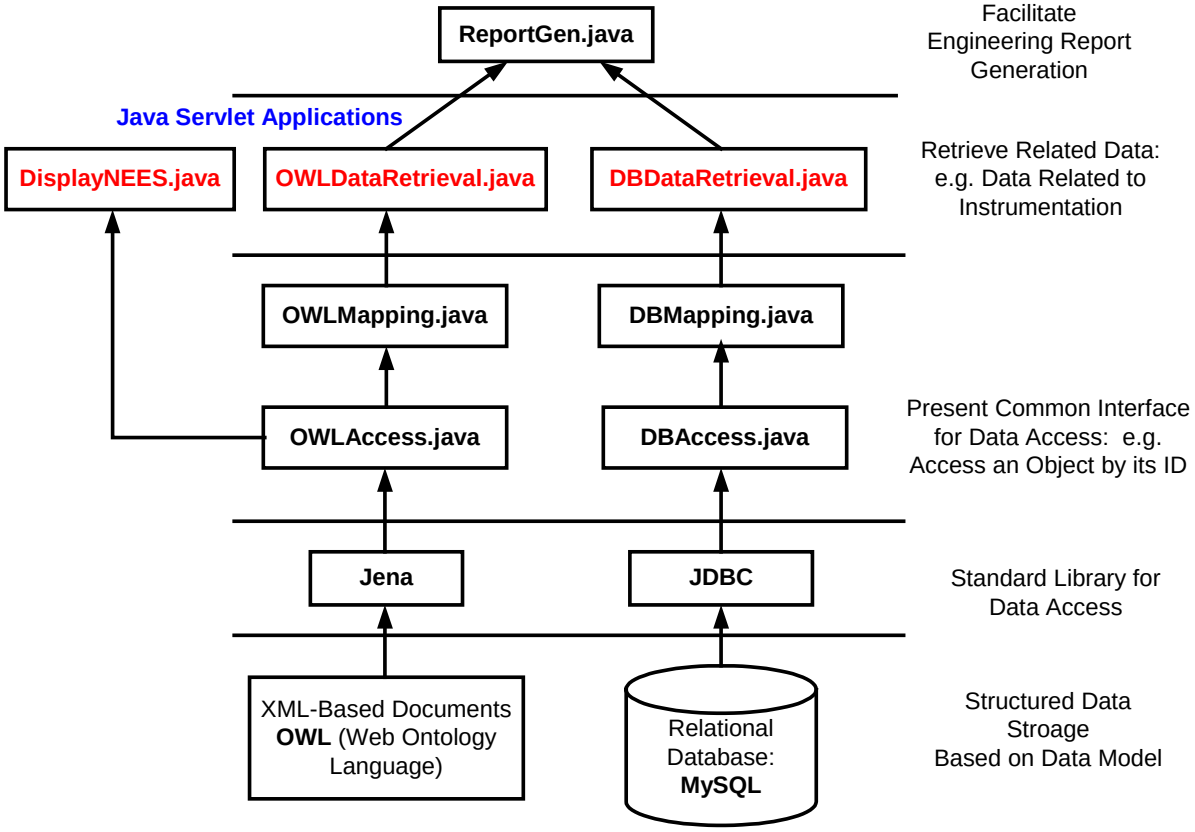

Figure 2: System Architecture

Currently, the data retrieval tools are implemented in Java. The relationships among the Java software modules are illustrated in Figure 2. The data and the metadata are saved either as OWL documents or in a relational database. Software tools, Jena and JDBC, are employed to access OWL files and MySQL databases respectively. A set of Java based tools are developed to retrieve and present the data to the users. For the data access layer, OWLAccess.java and DBAccess.java are implemented to present a set of common data access functions, such as accessing an object (in OWL files) or a

tuple (in MySQL) by its unique identification. OWLMapping.java and DBMapping.java are implemented to organize related data, such as assembling all the sensors setup information of a project. Data retrieval layer are built based on functions provided by data access layer. DisplayNEES.java is a data-browsing tool that allows users to navigate through the objects saved in OWL documents. Both OWLDataRetrieval.java and DBDataRetrieval.java are developed to allow the retrieval of related information and files of a project according to a pre-defined set of functions. These functions allow the retrieval of related data pertaining to a project. The data can be reorganized to a specific structure and to facilitate grouping the data for report generation purpose. ReportGen.java is the front-end program for users to specify the access of data from the different repositories.

3 Detailed Design

3.1 Data Browsing

A Project Viewer (DisplayNEES.java) is developed to browse the data and metadata related to a project on a client's web browser. The server side program is implemented using Java Servlet technology. The main methods defined in DisplayNEES.java are listed in Figure 3. This tool supports the display of all the fields of an object (such as Project, Event, SensorSetup, Specimen etc.). The fields include both the contents (such as the name, objective, startTime, and publications of a Project) and the links (such as the links from a Project to its Tasks). The contents are displayed on the browser and the links are displayed as hyperlinks that point to other objects. By following the hyperlinks, the data and the metadata pertaining to a project can be browsed and accessed.

```
/* Connect to the data repository and build an ontology model
   based on the OWL file */
loadModel();

/* Retrieve the name of all the projects in the ontology model */
listAllProjects(OntModel theModel);

/* Return all the slots of a particular class */
getClassSlots(String theClass);

/* Find the type of a particular property */
getPropertyType(Property oneProperty);

/* Find the slot value of a particular object */
getInstanceProperty(String theObject, String theSlot);

/* Retrieve all the slot values of an object */
getInstanceAllProperties(String theObject)

/* Display all the properties (slot values) of an object */
displayAnInstance(String theObject)
```

Figure 3: Class Interface of DisplayNEES

3.2 Data Retrieval

Researchers and engineers typically prefer to save experimental results and other project related information in a pre-defined file hierarchy. The file hierarchy serves as a way to categorize and manage project data. Different researchers and organizations prefer organizing the file hierarchy differently. For illustration, Figure 4(a) shows an example file hierarchy provided by Prof. Andrei Reinhorn of SUNY-Buffalo.

Data retrieval tools (OWLDataRetrieval.java and DBDataRetrieval.java) are developed to facilitate retrieving related data from the data repository and to present the data to the users according to the file hierarchy. As shown in Figure 4(b), a number of methods are defined, which directly correspond to the engineering file structure shown in Figure 4(a). For example, the method getScope() will retrieve data to be saved in the directory "Scope and general presentation".

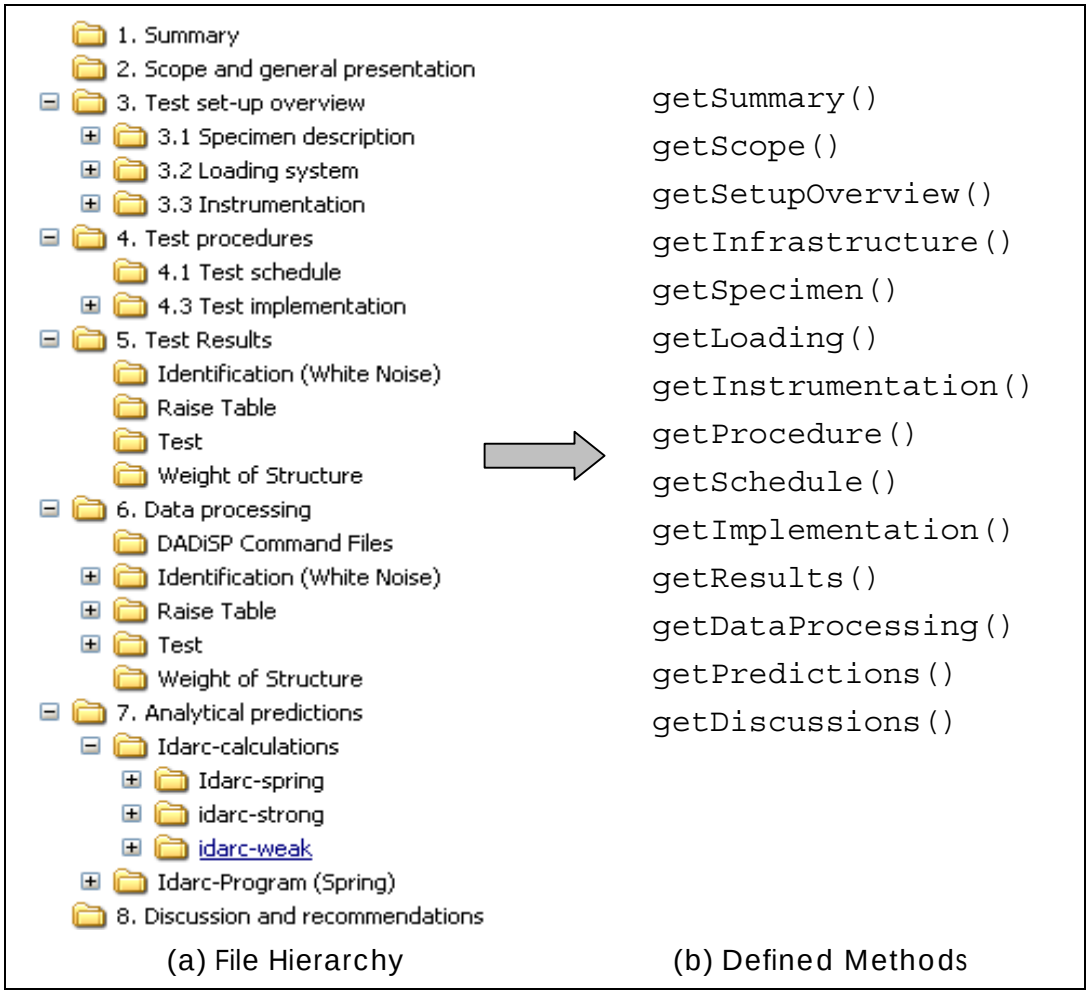

Figure 4: File Structure and the Corresponding Functions

3.3 Report Generator

Since the amount of data pertaining to a project can be voluminous, a user may not want to incorporate all the project data in an engineering report. A data selection process is still necessary in order to generate a concise and meaningful report. The Report Generator is implemented to help organizing project data according to the engineering file hierarchy. Once the file hierarchy is populated with data and files, the user can pick and choose relevant information to be incorporated in an engineering report.

The Report Generator (ReportGen.java) is implemented by using the data retrieval functions as schematically shown in Figure 4. Besides retrieving the project data from the data repository, the Report Generator also supports the scenario where different data repositories are employed to store project data. As shown in Figure 1, the project data can be retrieved from different data repositories using a single web-based interface.

4 User Interface Design

The user interface for all the data retrieval tools is a standard web browser. Since Java Servlet technology is used for implementing the server-side tools (in other words, all the functions and logics are on the server-side), there is no setup requirement on the client-side.

4.1 Interface of Data Browsing Tool

Figure 5 shows the main page of the project viewer showing a list of projects saved in the repository. When we click on a particular project, say miniMOST-1, details of the project are then shown on the browser, as illustrated in Figure 6. Links on the webpage can be followed to access details of other objects, which are rendered in an interface similar to Figure 6.

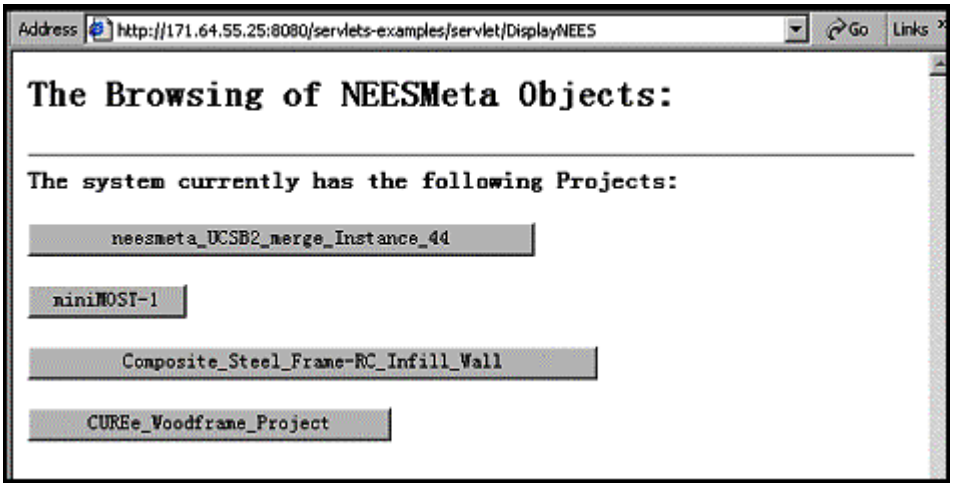

Figure 5: Main Interface of Data Browsing Tool

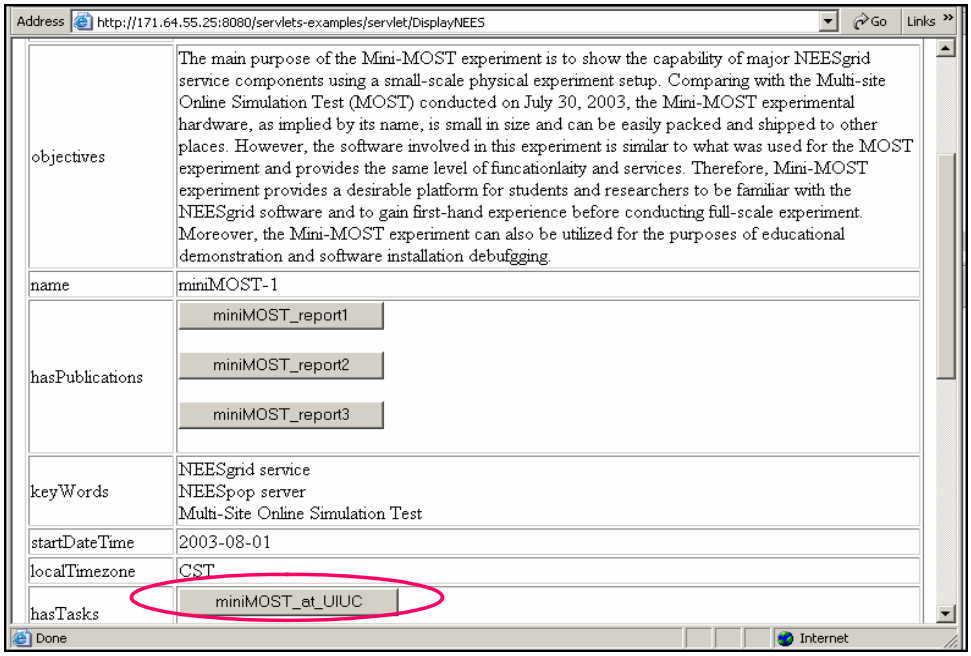

Figure 6: Detailed Display of an Object

4.2 Interface of Data Retrieval Tool

Figure 7 shows the main page of the data retrieval tool. A user can choose a project and the category of interest. Once the selection is processed, the retrieved data will be shown on the browser, as illustrated in Figure 8.

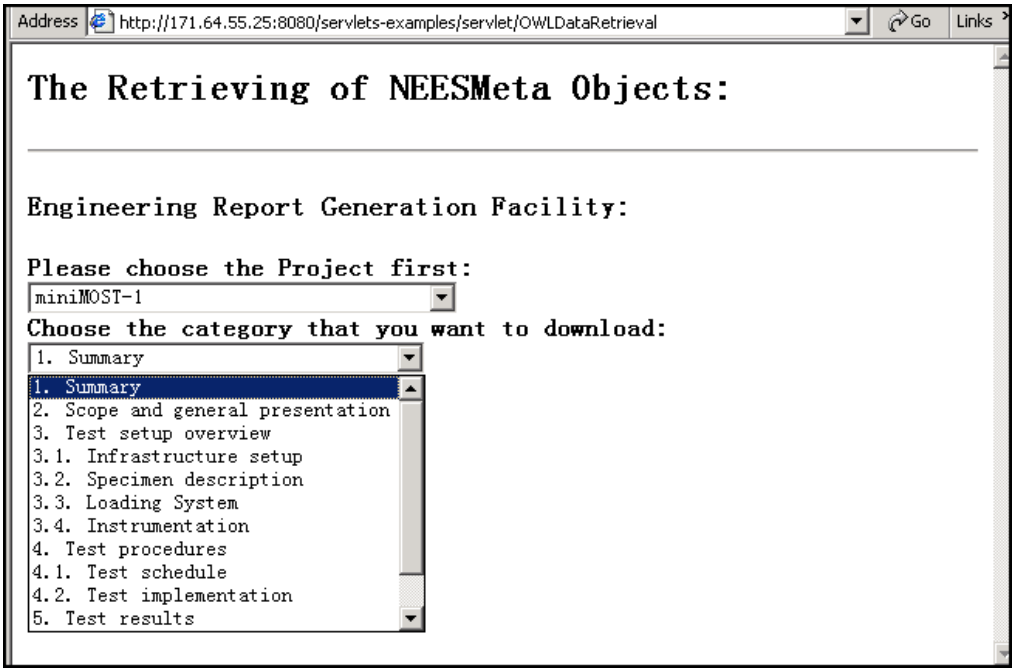

Figure 7: Main Interface of Data Retrieval Tool

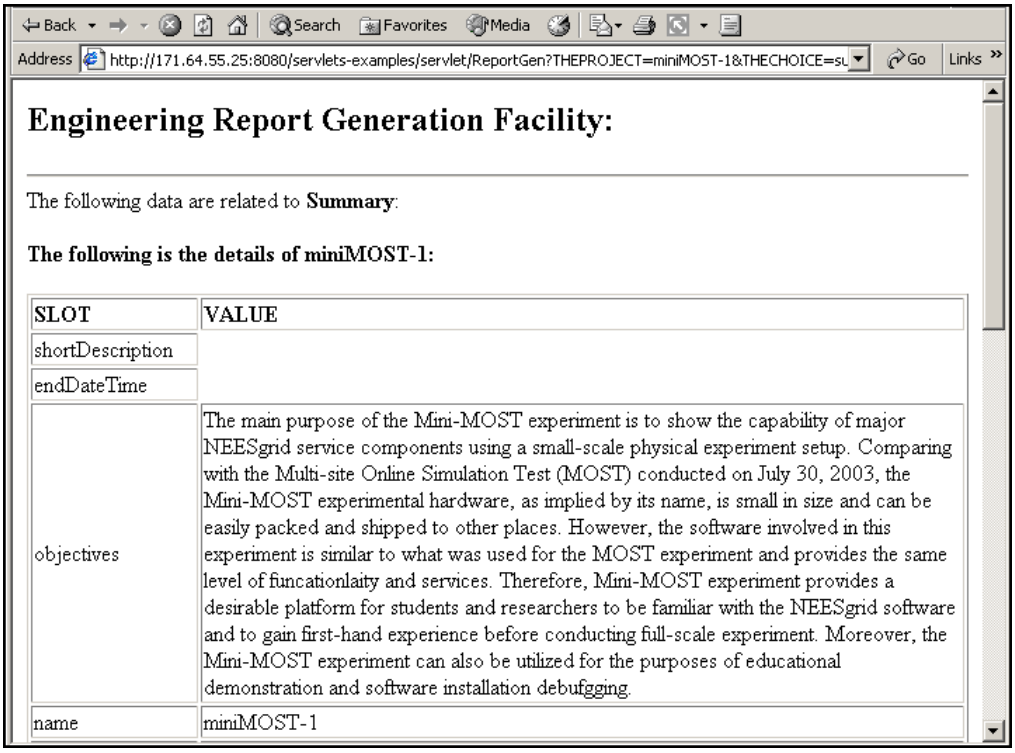

Figure 8: Retried Information Related to Summary

4.3 Interface of Report Generator

The web-based interface for the Report Generator is shown in Figure 9. The same interface is used for accessing the data saved in either OWL files repository or MySQL database. The left frame displays buttons which represent the categories as defined in the report file hierarchy. Upon clicking on a particular button, the right frame displays the content of that category, as illustrated in Figure 9.

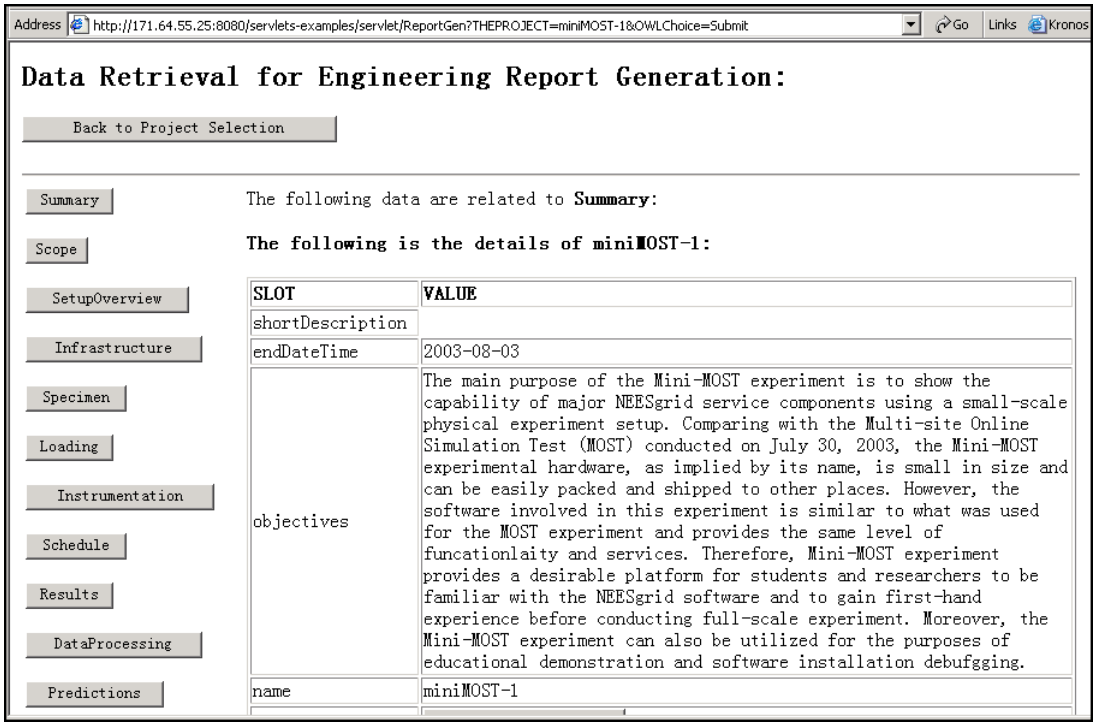

Figure 9: Main Interface of Report Generator

5 References

Jun Peng and Kincho H. Law (2004a). A Brief Review of Data Models for NEESgrid. NEESit Report TR-2004-01, http://it.nees.org/documentation/pdf/TR_2004_01.pdf.

Jun Peng and Kincho H. Law (2004b). Reference NEESgrid Data Model. NEESit Report TR-2004-40, from http://it.nees.org/documentation/pdf/TR-2004-40.pdf.

Tim Warnock (2005). Central Repository Design Specification. NEESit Report TR-2005-006, from http://it.nees.org/documentation/pdf/TR-2005-006.pdf.