Technical Report NEESgrid-2004-44

www.neesgrid.org

(Whitepaper Version: 1.0)

Last modified: July 12, 2004

Validity and Usability of the NEESgrid Reference Data Model

Jun Peng and Kincho H. Law[1]

[1]Department of Civil and Environmental Engineering, Stanford University, Stanford, CA 94305-4020

Feedback on this document should be directed to junpeng@stanford.edu

Acknowledgment: This work was supported primarily by the George E. Brown, Jr. Network for Earthquake Engineering Simulation (NEES) Program of the National Science Foundation under Award Number CMS-0117853.

Table of Contents

Table of Figures

1 Introduction

A reference data model for NEESgrid has been designed and developed [4,5]. The developed reference data model is based on the data requirements for shake table experiments. However, the model is of sufficient generality that major parts of the model can be modified and refined to support other types of experiments, such as centrifuge tests, pseudo-dynamic structural tests, and others. To ensure its usability, validation tests have been conducted by populating the data model with experimental data. Validation of the model will continue and refinements and updates will be incorporated as work progresses.

2 Validation Test

The usability of the reference data model has been tested with legacy experimental data. For the validation tests, Protégé [1] was employed as the interface to input experimental data and local file system was used as the storage medium.[1] For illustration purpose, this report focuses on the data set obtained from a Mini-MOST experiment [3].

2.1 Mini-MOST Experiment

The main purpose of the Mini-MOST experiment is to show the capability of the various NEESgrid service components using a small-scale physical experimental setup [3]. The Mini-MOST experimental hardware, as implied by its name, is small in size and can be easily packed and shipped to experimental sites. The Mini-MOST experiment provides a platform for students and researchers to become familiar with the NEESgrid software and to gain first-hand experience in using the NEESgrid services. The Mini-MOST experiment can also be utilized for educational demonstration and software installation debugging. For the validation test of the reference data model, the data were generated from a particular Mini-MOST test on February 28, 2004 at University of Illinois at Urbana-Champaign.

2.2 Inputting Experimental Data

Experimental data from the Mini-MOST experiment was ingested using Protégé [1] and saved as files in a local file system. Figure 1 shows loading an example project named miniMOST-1 into the system. Data are inputted using the slots (properties) as defined in the reference data model. If a slot is defined as primitive type, such as Integer, Real Number, Time, or String, etc., we can simply type in the value. If a slot is defined as Objects, then we can either choose a previously created object or create a new one. If a slot is defined as of type "URI" (which would normally refers to a file), we can save the particular file by entering the URI for the file location. Other types of objects, such as Task, EventGroups, Event, SensorSetup, InfrastructureSetup, Sensor, Specimen, and etc., can be created and inputted through an interface similar to the one shown in Figure 1. All the objects related to Mini-MOST experiment have been created and saved; the metadata and information about the data are

[1] Project Browser and data ingestion tools were under development and were not available for the validation tests.

saved as an OWL (Web Ontology Language) (http://www.w3.org/2001/sw/WebOnt/) file. Other experimental data, such as specimen photos and sensor readings, can be stored in a file on a web server with its URI saved in the OWL file.

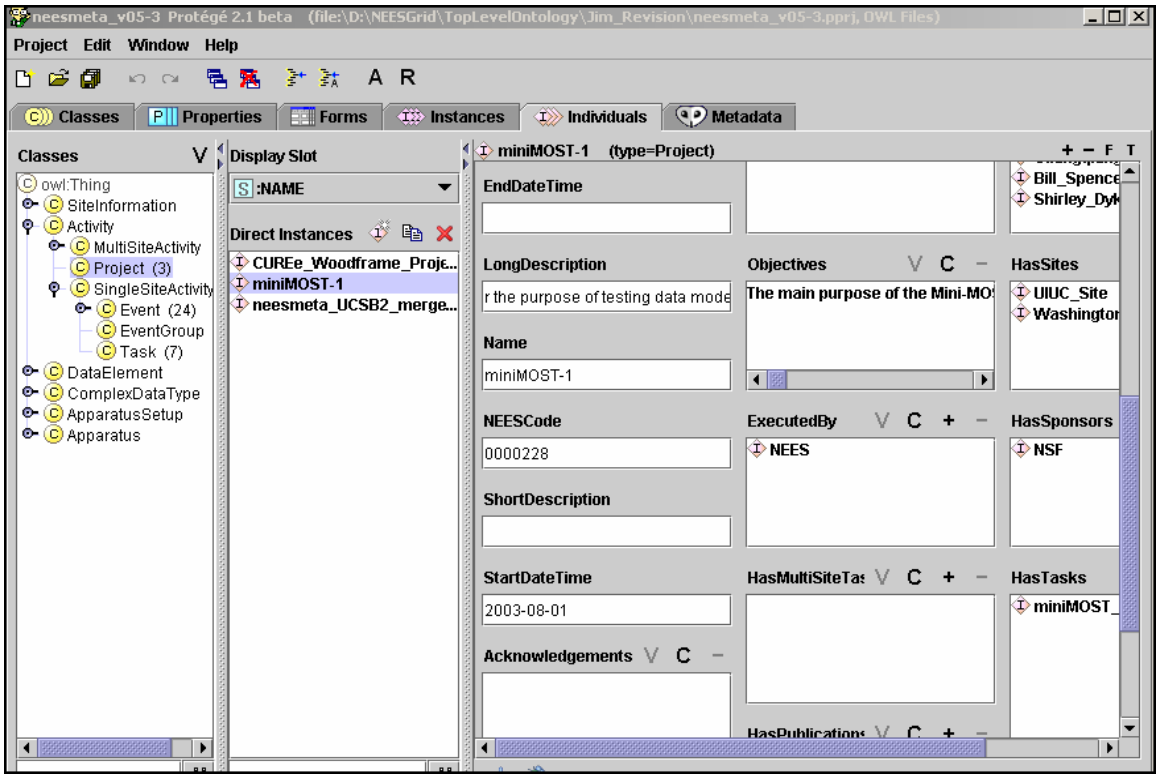


Figure 1 – Using Protégé to Input Mini-MOST Experiment Data

2.3 Browsing Experimental Data

For validation purpose, we implemented a project viewer to retrieve the saved data and to view the data on a web browser according to the data model. The program is implemented using Java Servlet technology (http://java.sun.com/products/servlet/), and the parsing of the OWL file is handled by using Jena [2]. Figure 2 shows the front page of the project viewer with a list of saved projects. When we click on a particular project, say miniMOST-1, the details of the project will be shown on the browser, as illustrated in Figure 3.

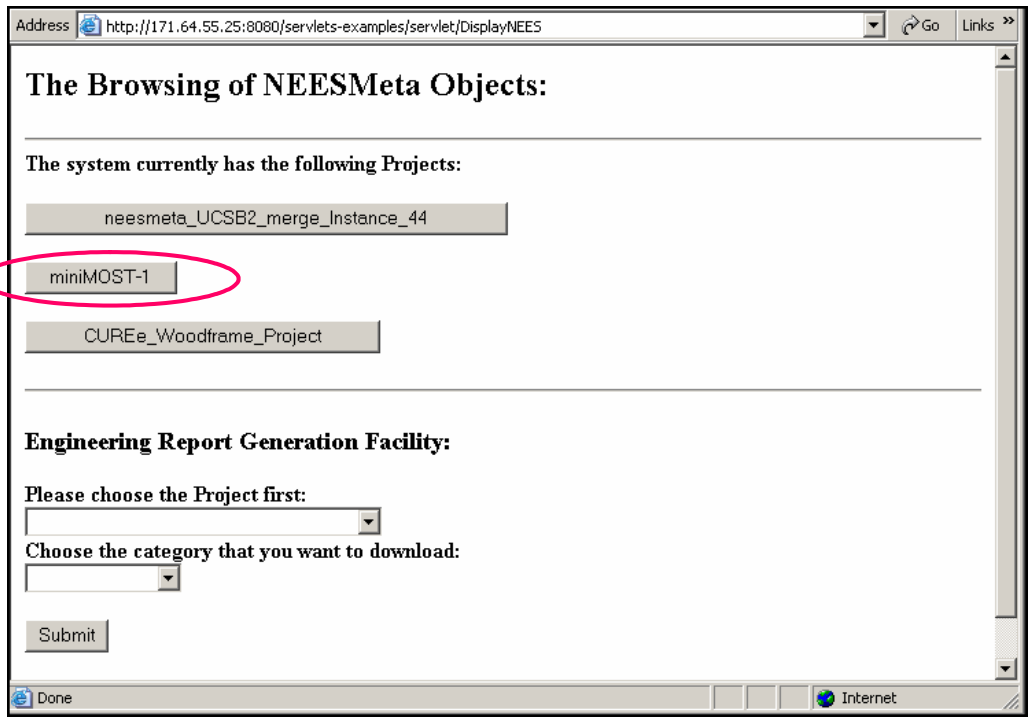


Figure 2 – The Front Page of the Project Viewer

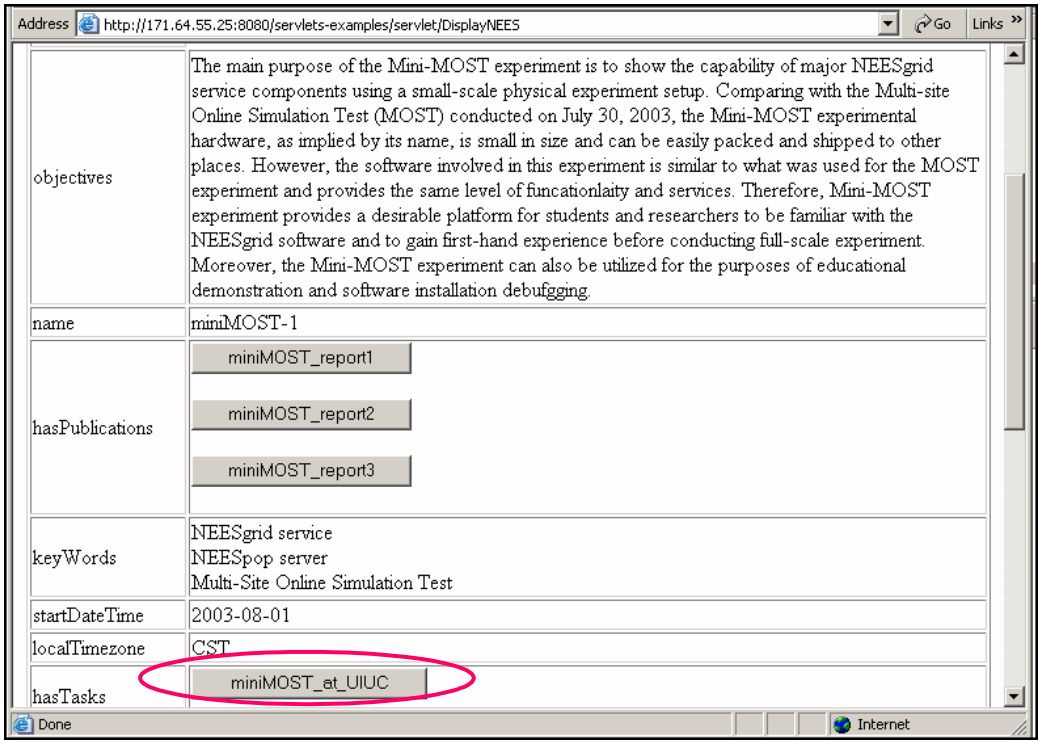


Figure 3 – Detailed Display of the Project MiniMOST-1

As defined in the reference data model, a Project is a collection (organized group) of Tasks designed to achieve specific goals and objectives. Following the model, we can navigate and access all the Tasks that belong to the Project. Figure 4 shows the details of a particular Task named miniMOST_at_UIUC. One property (or a slot) of a Task object is InfrastructureSetup, which models the assembly and arrangement of the PrimaryEquipment used for a specific Task. We can access the details of the InfrastructureSetup object by clicking on the highlighted button as shown in Figure 4.

Figure 5 presents the details of the InfrastructureSetup, which essentially is a collection of texts, documents (in the format of Word, PDF, Excel, etc.), figures and drawings stored as files. Files are saved in a web server and their URIs are saved as metadata. The files can be dynamically downloaded and shown on a web browser, as illustrated in Figure 6.

Each Task in a project may contain one or more EventGroups. The EventGroup object can be accessed by clicking on the highlighted button shown in Figure 7. The details of a particular EventGroup object named miniMOST_UIUC_EventGroup_2004 are presented in Figure 8.

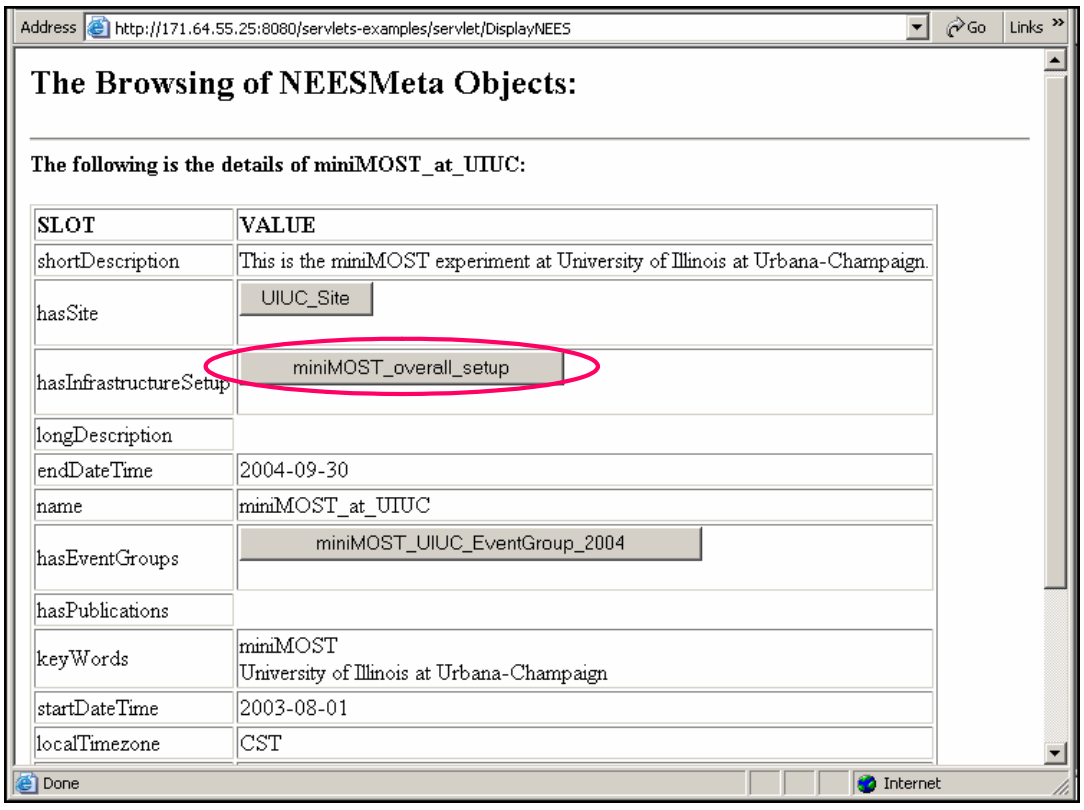


Figure 4 – Detailed Display of the Task miniMOST_at_UIUC

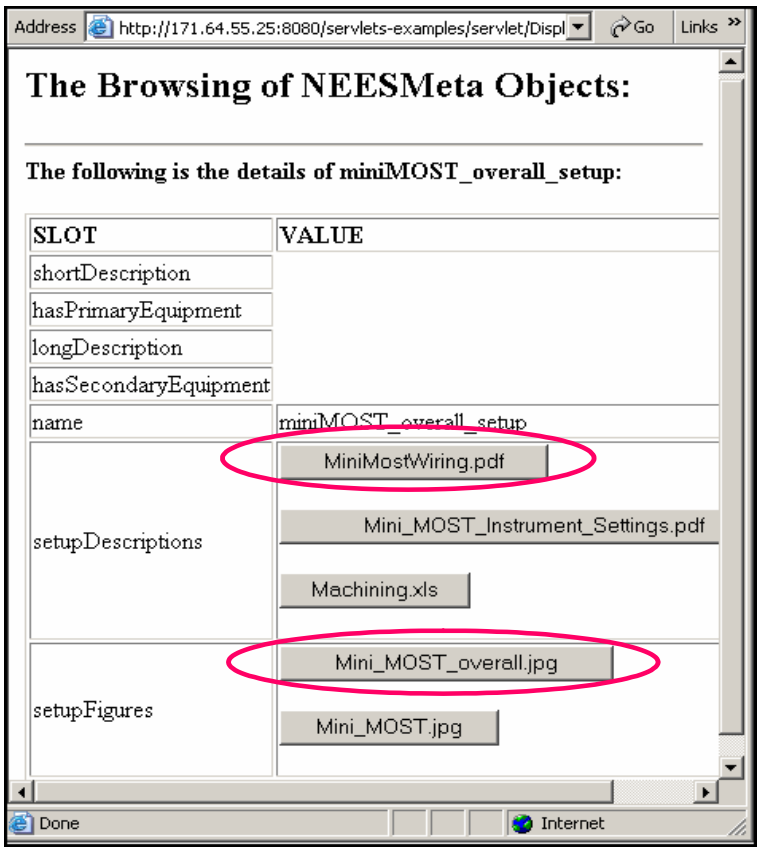
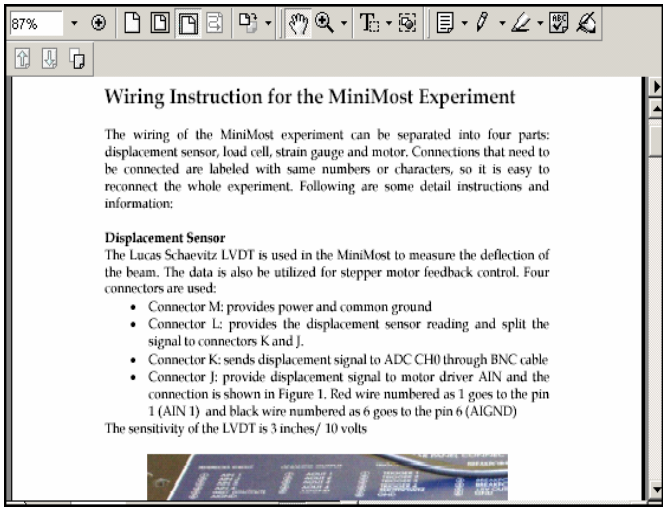
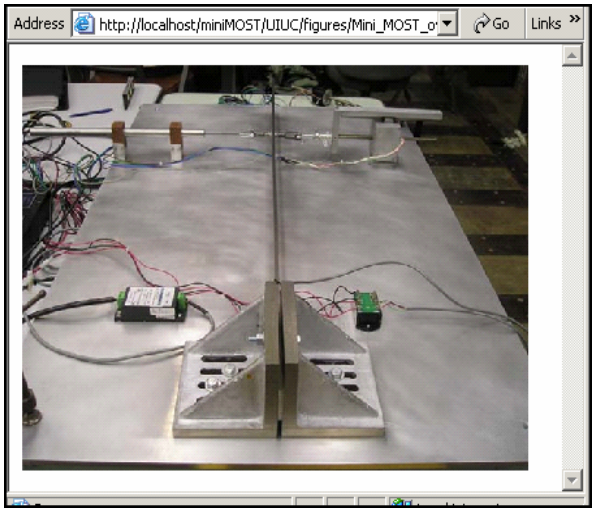


Figure 5 – Detailed Display of the InfrastructureSetup



(a) MiniMostWiring.pdf



(b) Mini_MOST_overall.jpg

Figure 6 – Access of Files Representing the InfrastructureSetup

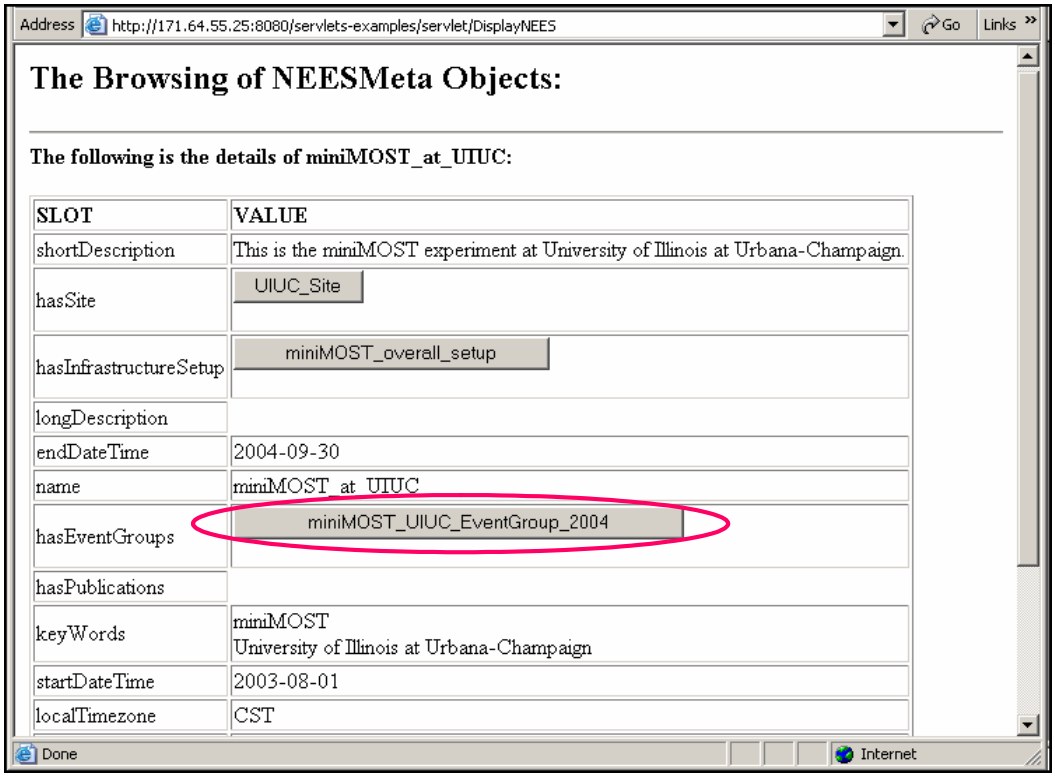


Figure 7 – Detailed Display of the Task miniMOST_at_UIUC

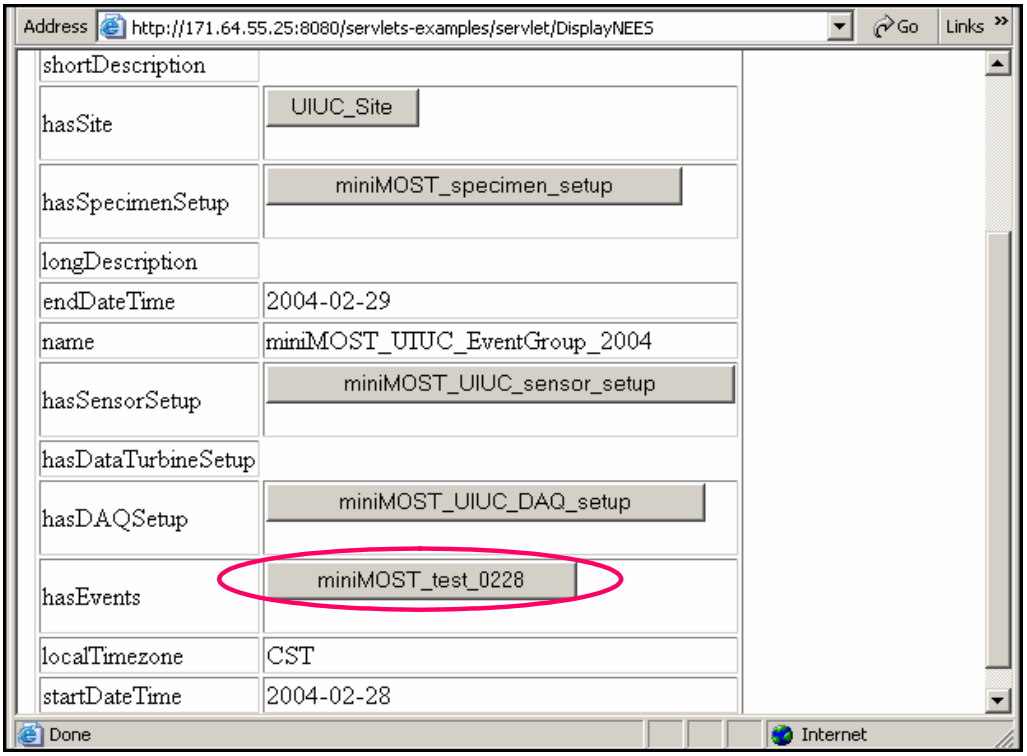


Figure 8 – Detailed Display of the EventGroup

An EventGroup is defined as a collection of Events, each of which can be accessed from the EventGroup object. The details of an Event named miniMOST_test_0228 are shown in Figure 9. An Event, which is the atomic level of Activity, refers to each single run of an experiment or a simulation. Experimental results, such as SensorReading, can be accessed from an Event object, as shown in Figure 10.

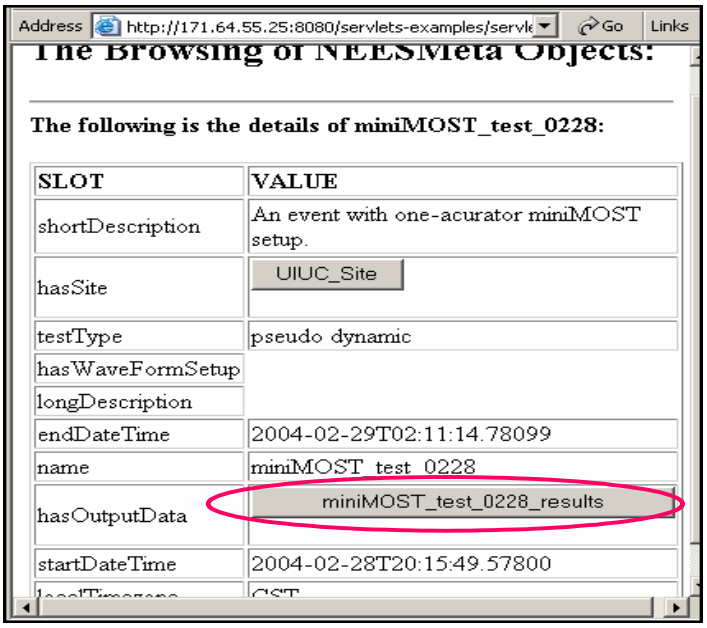


Figure 9 – Detailed Display of the Event miniMOST_test_0228

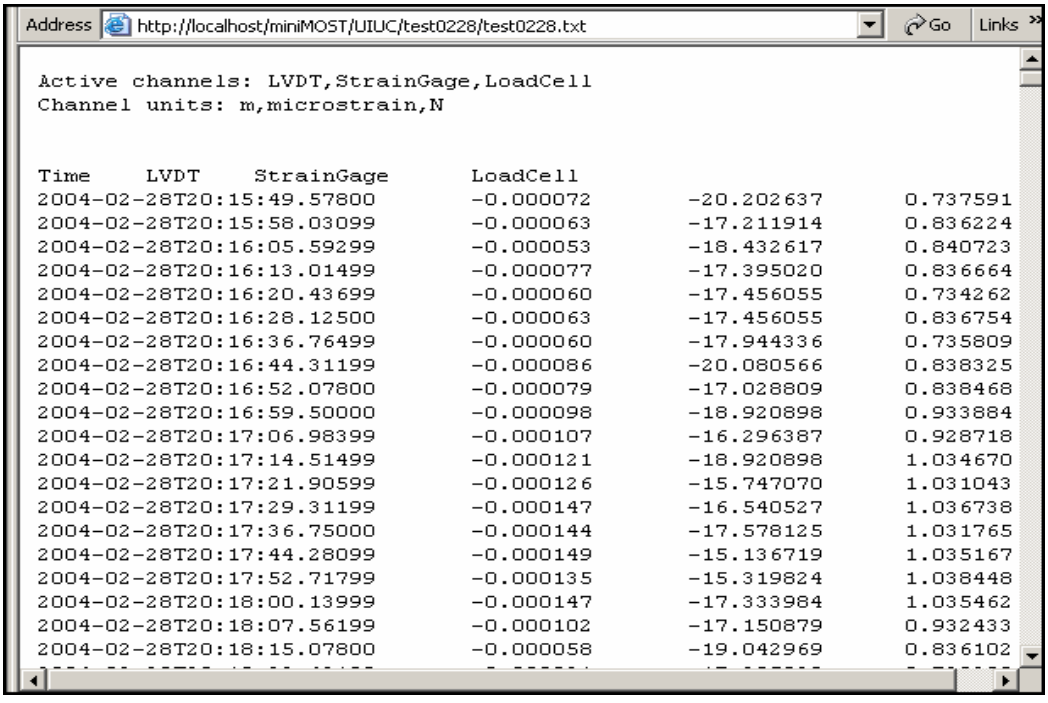


Figure 10 – Access of SensorReading for the Event miniMOST_test_0228

The EventGroup object also contains the objects of SpecimenSetup, SensorSetup, and DAQSetup. Figure 11 shows the details of the SensorSetup object, which belongs to the EventGroup named miniMOST_UIUC_EventGroup_2004. Again, the setup is described in texts, documents, drawings and picture files. Each file can be accessed by simply following the URI for the file. For example, Figure 12 shows a photo for the setup of a LVDT sensor.

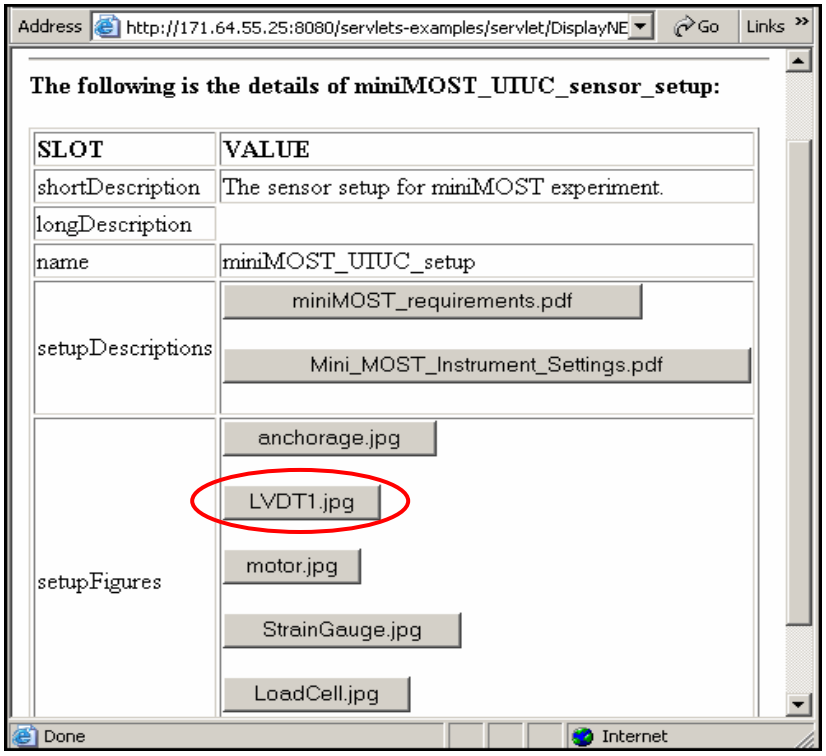


Figure 11 – Detailed Display of the SensorSetup

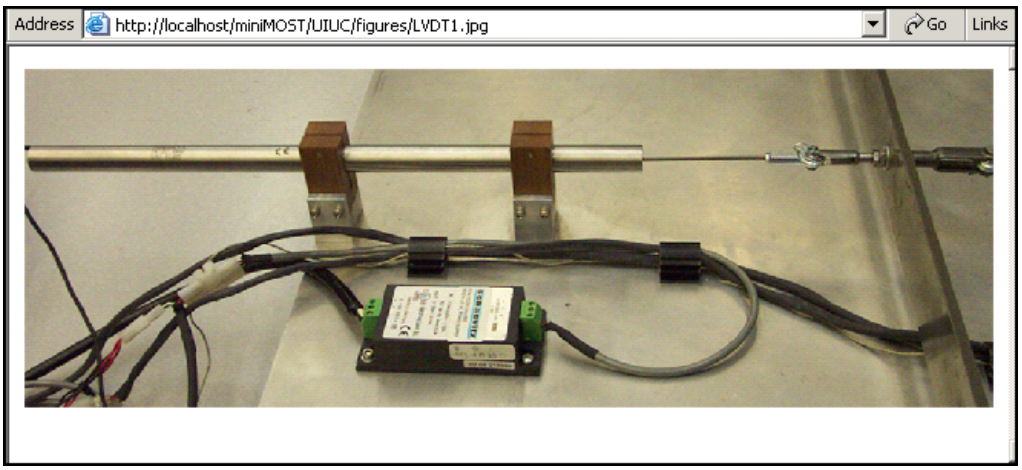


Figure 12 – Access of Photo for the LVDT

3 Summary and Discussion

To validate the reference data model, we have populated the model with the mini-MOST experimental data provided by UIUC. This validation process helps evaluate the completeness, flexibility and usability of the data model. The usability test has demonstrated that the data model is sufficiently comprehensive to save and organize all the mini-MOST data. In addition, as the experimental data are organized according to the data model, browsing and accessing them are fairly intuitive and straightforward. Efforts will continue to validate, evaluate and refine the reference data model using other experimental projects and data.

References

1. J. Gennari, M. A. Musen, R. W. Fergerson, W. E. Grosso, M. Crubézy, H. Eriksson, N. F. Noy, and S. W. Tu. *The Evolution of Protégé: An Environment for Knowledge-Based Systems Development*, Stanford Medical Informatics, Stanford University, 2002. (http://smi.stanford.edu/pubs/SMI_Abstracts/SMI-2002-0943.html)
2. B. McBride, D. Boothby, and C. Dollin. *An Introduction to RDF and the Jena RDF API*, 2004. (http://jena.sourceforge.net/tutorial/RDF_API/index.html).
3. N. Nakata, G. Yang, and B. F. Spencer. System Requirements for Mini-MOST Experiment, NEESgrid Technical Report, 2004. (http://www.neesgrid.org/mini-most/Mini_MOST_requirements_revised3.pdf).
4. J. Peng and K. H. Law. *A Brief Review of Data Models for NEESgrid*, Technical Report NEESgrid-2004-01, 2004. (http://www.neesgrid.org/documents/TR_2004_01.pdf)
5. J. Peng, K. H. Law, and G. Pekcan. *Reference NEESgrid Data Model for Shake Table Experiment*, NEESgrid Technical Report, 2004.