13th World Conference on Earthquake Engineering
Vancouver, B.C., Canada
August 1-6, 2004
Paper No. 4001

OVERVIEW OF NEES DATA AND METADATA MODELS

Jean-Pierre Bardet[1], Kincho H. Law[2], Jun Peng[2], and Jennifer Swift[1]

SUMMARY

NEESgrid, the system integration of the NEES research collaboratory, uses recent advances in Information Technologies for modeling the data sets and experimental processes generated at the NEES testing sites. NEESgrid describes experimental results and processes using object-oriented data models expressed in NEESML, the NEES Metadata Language, which is a subset of RDF (Resource Descriptor Framework). At the present, data models and data dictionaries are being developed for various fields of earthquake engineering using Protégé-2.0, based on user surveys, usage scenarios and existing data sets.

INTRODUCTION

The George E. Brown Jr. Network for Earthquake Engineering Simulation (NEES) is a pioneering endeavor in earthquake engineering. Funded by the US National Science Foundation, NEES is a research collaboratory that connects, through a high performance network, several types of earthquake engineering testing sites located at different places in the United States. The NEES testing sites include shake tables, centrifuges, tsunami/wave tanks, large-scale laboratory experimentation systems, and field experimentation and monitoring installations. All these distributed testing facilities are integrated into a research collaboratory by the system integration of NEES (i.e., NEESgrid), which offers a curated data repository, enables tele-observation and tele-operation, and has capabilities for distributed computer simulation.

Before NEES, experimental results and processes in earthquake engineering were rarely documented in great detail as those were mostly carried out by isolated investigators at single institutions, and were partially published in hardcopy reports and papers in technical journals and proceedings. There have been few examples in earthquake engineering which have emphasized the importance of preserving the information generated by collaborative research projects, e.g., ROSRINE (Resolution Of Site Response Issues from the Northridge Earthquake) [1]; VELACS (Verification of Liquefaction Analysis using Centrifuge Studies) [2]; COSMOS GDM (Geotechnical Data Model) [3]; SAC steel project [4]; and CUREE-Kajima woodframe project [5]. These past collaborative works in earthquake engineering have illustrated the complexity and dedication inherent to the construction of data and metadata models.

---

[1] Civil Engineering Department, University of Southern California, Los Angeles, CA 90089-2531
[2] Department of Civil and Environmental Engineering, Stanford University, Stanford, CA 94305-4020

Within the NEES collaboratory, data models are critical to exchange the experimental and computational information generated at the distributed testing sites and computing facilities. Data models are needed not only for ingesting data into the data repository, but also to understand and analyze the data subsets queried from the data repository. NEES prompted earthquake engineers to revisit the ways they have stored and published information, and to learn from experiences from other scientific fields. The objective of this paper is to summarize the experiences and strategy in developing the NEES data/metadata models. Following the introduction, the second section reviews the approaches available for modeling data, and the third section summarizes a few examples of data models relevant to NEES. The fourth section introduces the approach adopted by NEESgrid and presents the most recent findings on NEES data models.

REVIEW ON DATA MODELING

Figure 1 illustrates the overall usage scenario of the data sets and experimental processes with the NEES collaboratory.

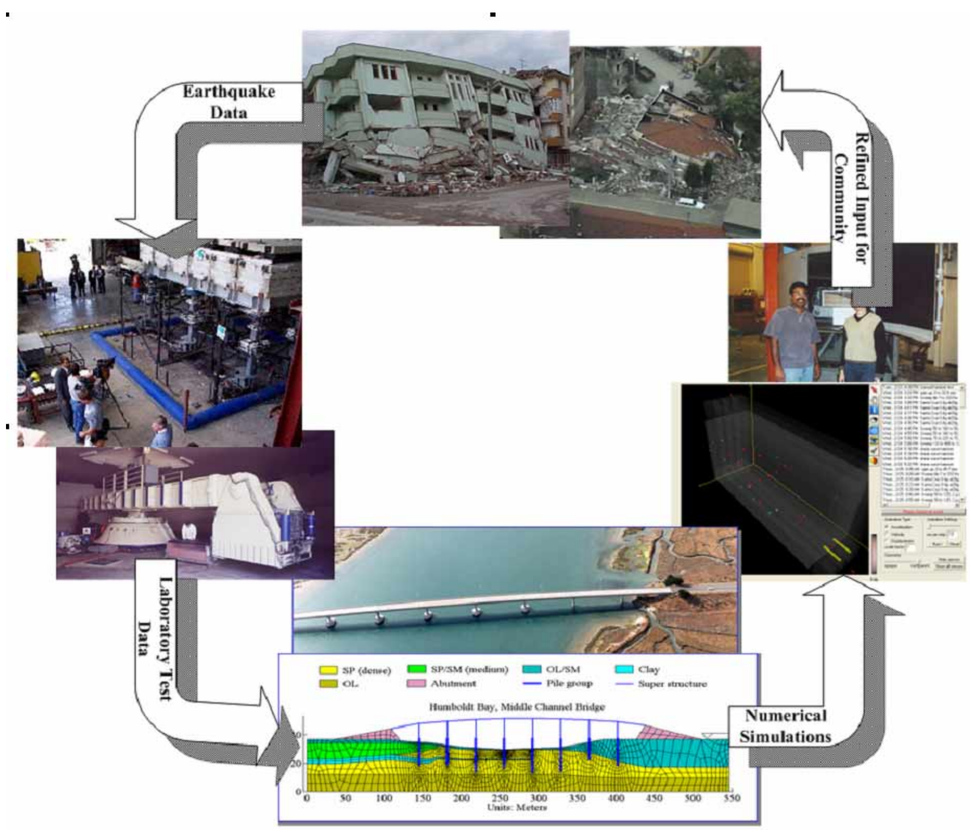

Figure 1. Usage scenario and data life cycle in the NEES collaboratory (photos courtesy of S. Mahin, B. Kutter and G. Fenves).

Researchers and engineers simulate critical problems in earthquake engineering using physical models in the laboratory and in the field, and examine the response of these physical models to earthquake-like loading conditions. With the rapid advancements of sensing techniques (e.g., MEMS), the physical models become increasing sophisticated, and require using several testing facilities together. For instance, landslide tsunamis, which have been identified as high-risk natural hazards after earthquakes [6], can only be studied by combining the resources offered by the NEES geotechnical centrifuges and tsunami wave tanks. Experiments, which required only tens of sensors before NEES, are envisioned to have thousands of sensors. These numerous sensors yield a voluminous amount of experimental data to be analyzed using distributed computed resources. A more complete vision for the future of physical and computational modeling within the NEES collaboratory can be found in the proceedings of a series of NEES user requirement workshops [e.g., 7]. In view of its emerging challenges in physical modeling, the NEES collaboratory needs definitely a new generation of data models. Ideally, the data models should be independent of hardware/software platforms. Data models can be designed using many data modeling tools and structures (e.g., relational, object-oriented model, and hierarchical structures). This section briefly reviews some of the methods and definitions used in data modeling.

Relational Models

Relational models are commonly used to store large experimental data sets and to extract subsets of information relevant to particular problems through queries [e.g., 8, 9]. Relational models are based on well established principles and are supported by many computer packages, unfortunately too numerous to be listed herein. Relational models store data about particular subjects in tables with fields (i.e., columns) and link data in different tables using keys. Relational model data are usually searched and processed using Structured Query Language (SQL), a standard language common to all SQL-compliant systems. SQL allows extraction of subsets of information from one or several tables and reporting in tables that can be understood by non programmers. Relational models are however recognized to have limitations [e.g., 8] when encountering the complex data structures of modern computer applications, e.g., Geographic Information Systems (GIS), and Computer Aided Design (CAD).

Entity-Relationship (E-R) Model

Widely adopted for relational models, the Entity Relationship (E-R) model organizes data graphically into entities and defines the relationships between the entities [e.g., 8, 9, 10]. The principal components of E-R models are as follows:

- **Entity sets** are the "things" (real or abstract) in which specific data will be stored; for example, entities may include project, event, specimen, researcher, sensor, measurement, and time. Specific occurrences of an entity are called **instances**, e.g. *John Smith* may be an instance of the entity *researcher*.

- **Attributes** are properties that describe an entity, e.g. project title, description, keywords, and objectives are all attributes of a project entity. An attribute or a combination of attributes that uniquely identifies one and only one instance of an entity is called a **primary key** or **identifier**. For example, projectID uniquely identifies project.

- **Relationships** are the associations among two or more entities, e.g. a project *is made of* events, and sensors *are attached to* a specimen. **Cardinality** defines the number of occurrences of one entity, e.g. a project may have multiple events, which may include pre-experimental events, actual experiments, computer simulations, and post-experimental events.

Due to its graphical nature, an E-R model is easy to understand and can be mapped directly to relational models. Often, domain (attribute) and referential integrity constraints are included when defining the entities and relationships to ensure consistency and correctness for the relational model.

Object-Oriented Model

Object-oriented models [11, 12] represent information as objects, which can be any kind of (real or abstract) entities. An object encapsulates related data as attributes, and connects with other related objects through some relationships. The relationship types commonly used may include *classification, association, aggregation* and *generalization*. These relationship types may in turn impose certain "object-oriented" features (e.g., inheritance) and integrity constraints. Object-oriented data model can be represented using the object definition language (ODL) and the unified modeling language (UML). ODL [12] is a proposed standard language for specifying the structure of object-oriented databases. ODL allows object-oriented models to be written and translated directly into object-oriented databases. UML [13, 14] is the industry-standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems. Although UML is primarily designed for designing object-oriented software, it can also be used to design object-oriented data models.

Object-Relational Models

Object-relational models [e.g., 8] offer a compromise between relational and object-oriented models, and are preferred by several major vendors (e.g., DB2 and Oracle 8i) as they combine the efficiency of relational models and flexibility of object-oriented models.

XML-based Data Model

The eXtensible Markup Language (XML) (http://www.w3.org/XML/) has become a standard for exchanging data over the internet. XML is a meta-markup language that consists of a set of rules for creating semantic tags used to describe data [15]. An XML element is made up of a start tag, an end tag, and content in between. The start and end tags describe the content within the tags, which is considered the value of the element. In addition to tags and values, attributes are provided to annotate elements. XML has wide acceptance from the computer industry, from application and database vendors such as Microsoft, AutoCAD, IBM and Oracle. XML has also been widely used for defining data models. The key advantages of XML are its object-oriented structure and readability extensibility.

XML Schemas (http://www.w3.org/XML/Schema) provide a means for defining the structure, content and semantics of data models. XML Schemas consists of components such as type definitions and element declarations. They can be used to assess the validity of well-formed attributes, and may specify default values for attributes and attribute types. The schema-validity assessment checks the constraints on attributes, and can be used to model the constraints imposed on data models.

Web-ontology and RDF

Ontologies define a common vocabulary for researchers who need to share information in a technical domain [16, 17, 18]. It includes machine-interpretable definitions of basic concepts in the domain and relations among them. Originated from the field of Artificial-Intelligence, ontologies have become common on the World-Wide Web to categorize information on large Web sites (e.g., Yahoo!) and products for sale (e.g., Amazon.com). Many disciplines now develop standardized ontologies which domain experts can use to share and annotate information in their fields. Medicine, for example, has produced large, standardized, and structured vocabularies such as SNOMED [19, 20]. The OWL Web Ontology Language [21] is designed for the applications that process the content of information instead of just presenting information to humans. OWL facilitates greater machine interpretability of Web content than that supported by XML and RDF by providing additional vocabulary and a formal semantics.

RDF

The Resource Description Framework (RDF), which is developed by the World-Wide Web Consortium (W3C), provides the foundation for metadata interoperability across different resource description communities [22, 23]. One of the major obstacles facing the resource description community is the multiplicity of incompatible standards for metadata syntax and schema definition languages. RDF (http://www.w3.org/RDF/) provides a solution to these problems via a Syntax and Schema specification. The RDF key concepts are graph data model, URI (Uniform Resource Identifier)-based vocabulary and node identification, data types, literals, XML serialization syntax, expression of simple facts, and entailment. RDF extends the XML model and syntax to be specific for describing resources. RDF utilizes the XML Namespaces, which point to URIs, and allows RDF to scope and uniquely identify a set of properties. This set of properties, called a schema, can be accessed at the URI identified by the namespace.

Data Modeling Tools

Many data modeling and software design tools [24] can be used to design data models for particular applications. For instance, Protégé-2.0 [25, 26, 27] has a graphical user interface (GUI) that facilitates the development of data models. Protégé-2.0 is open-source software, the capabilities of which are enhanced by various plug-ins. Some plug-ins export Protégé-2.0 data models into different formats, including UML, OWL, XML Schema, and RDF. The graphical interface of Protégé-2.0 enables description of particular objects using classes and sub-classes and relates them through various types of relationships. Figure 2 shows an example of the Protégé-2.0 GUI, with classes and subclasses shown in the left window. The lower right window shows detailed attributes (slots) of these classes. Protégé-2.0 allows design of entry forms to input data, and to query subsets of data. In its present version, Protégé-2.0 is intended for designing data models, which can be tested for relatively small data sets. It is not intended to operate on several terabytes of information.

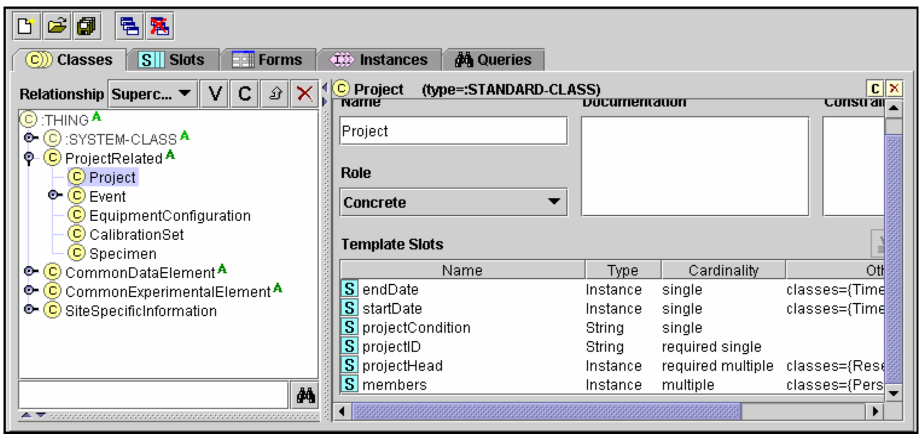

Figure 2. Protégé-2.0 Interface.

DATA MODELS RELEVANT TO NEES

Data models, if they are to support the NEES collaboratory, must include different types of information, encompassing project management, experimental models, sensors, persons, raw data, and publications. The previous section mentioned several approaches for modeling data in sciences and medicine. These data models, even if they originally appear remotely related to earthquake engineering, present commonalities advantageous to the development of NEES data models. The following section examines some of these models and their relevance to the NEES data/metadata efforts.

Ontology of Science

The Ontology of Science [28] was designed for modeling scientific events and educational events, e.g., a scientific conference, a research project, or a software development project. The ontology has a high-level project entity. As shown in Figure 3, the project has relationships with other entities, including people, organizations, products and events. The project has a start date and an end date, which can be used to calculate the project duration. Besides the high-level modeling, the Ontology of Science provides detailed modeling of several common elements, such as people, scientific documents location, and time. The ontology offers an organizational representation useful to organize NEES projects.

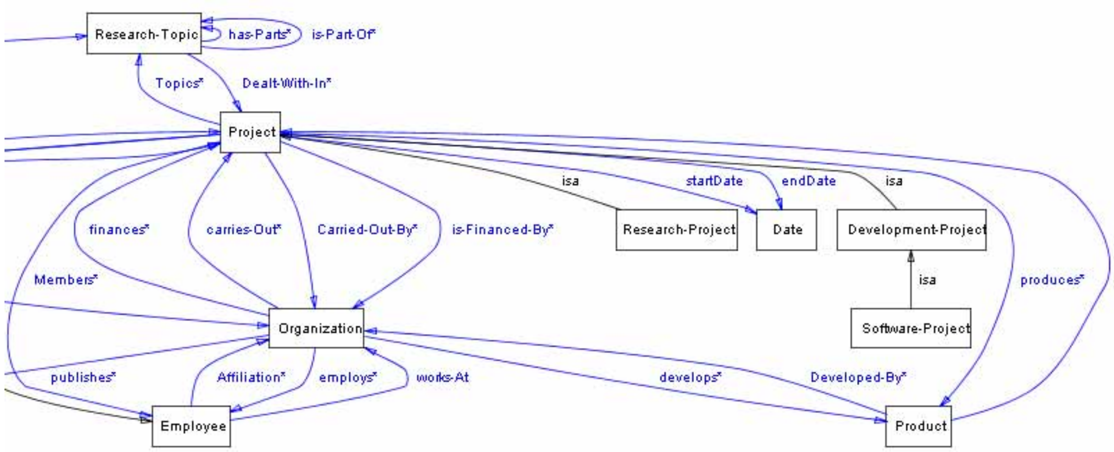

Figure 3. Project Model of the Science Ontology

Sensor ML

SensorML [29, 30] is an XML-based Sensor Model Language which was developed for describing the geometric, dynamic, and radiometric properties of dynamic sensors of earth sensing satellites. Sponsored by the international Committee for Earth Observing Satellites (CEOS) and the OpenGIS Consortium, SensorML is under review as part of ISO TC211 Projects [31]. As shown in Figure 4, SensorML defines the geometric, dynamic, and observational characteristics of a sensor using an XML schema. The root for all SensorML documents is *Sensor,* which represents a device for the measurement of physical quantities. Key components of a sensor modeling include sensor identification, sensor location, constraints, platform attached by the sensor, coordinate reference system, sensor description, and measurement characteristics.

An important concept of SensorML is SensorGroup. A SensorGroup can be of two types, sensor package and sensor array. A sensor package is composed of multiple sensors that operate together to generate a collective observation or related group of observations. For example, a collection of sensors can be used in a combined fashion to create a sensor that measures wave velocity and direction. A sensor array is a set of sensors of the same type at different locations. These locations may be within a single sensor frame, a different location on a single mount, or on different platforms. A sensor array produces observations that are used to build a spatial coverage. SensorML provides a good prototype and a rich model for describing sensor information. It is particularly useful for in-situ experimental applications involving spatially distributed information.

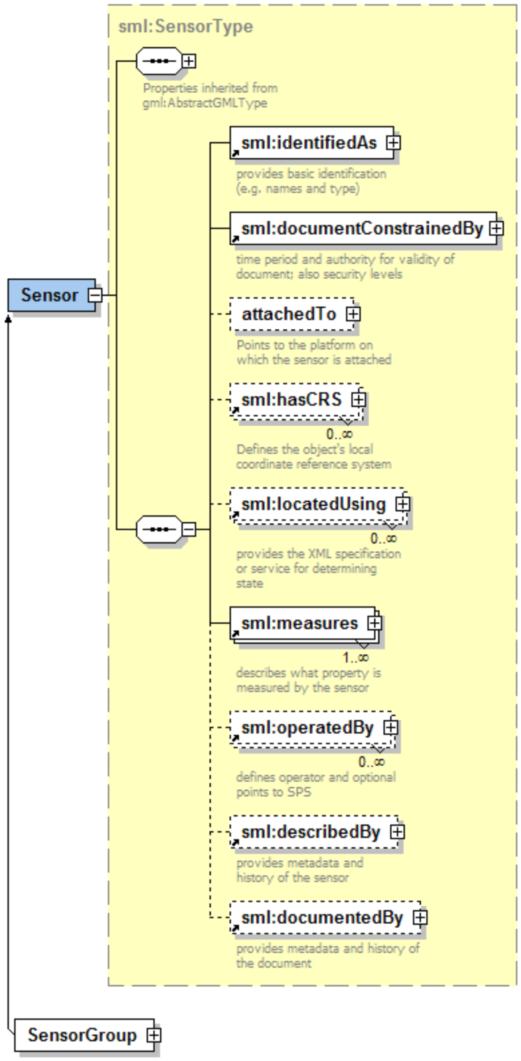

Figure 4. Schema diagram showing the highest level of the SensorML document structure [29].

Berkeley CUREE/ Kajima

In the context of the CUREE/Kajima research project, a framework was recently developed at Berkeley for integrating and visualizing structural engineering data [32]. This framework gathers, classifies and integrates structural data collected in and around a structure, and enables an effective visualization and fusion of the data defining the state of a structure. Presently, the model focuses on the collection of observed data. The model describes comprehensively collector and data type. As shown in Figure 5, the abstract class *Collector* is a general object that collects or generates data about a structure. A Collector can be any type of sensor, a camera, numerical analysis, or even a person. Each collector type is capable of collecting different types of data and handling different types of input and output data. The observed or generated data may include numerical data, descriptive textual data, or visual and graphical data (e.g., visual imagery from a camera, hand sketch, and drawings). The Berkeley CUREE/Kajima model represents well the different types of sensors commonly used for structural monitoring. Individual models of sensors are defined as separate objects, which have special calibration information. As shown in Figure 5, different types of sensors, e.g., accelerometers, strain gages, thermocouple, are modeled as subclasses of Sensor.

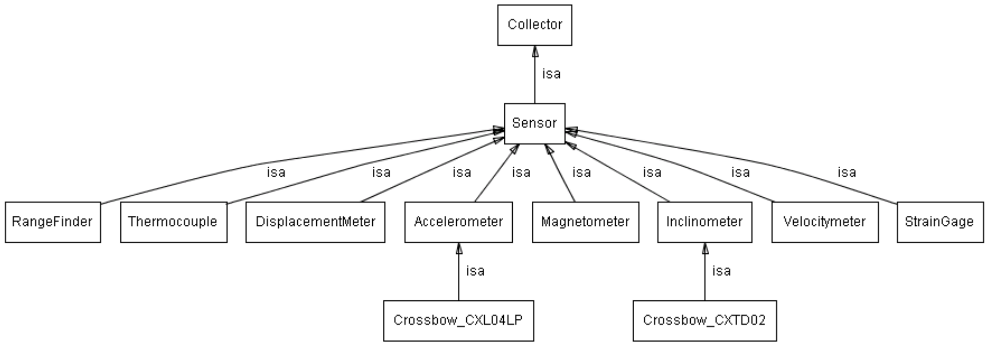

Figure 5. Sensor Model of the Berkeley CUREE/Kajima

Solid Geometry Models

There are several universal models for describing solid geometries (e.g., structures and components of buildings) and exchanging data between Computer Aided Design (CAD) systems. The CIMSteel Integration Standards (CIS/2) (http://cic.nist.gov/vrml/cis2.html) is the product model and electronic data exchange format for structural steel project information [33]. The CIS/2 standard, which is based on STEP (STandard for the Exchange of Product Data [34]), describes structures and engineering information, testing procedures and industry specific information. CIS/2 has been implemented for describing steel framed structures, from nuts and bolts to materials, loads to frames and assemblies. Adopted by the American Institute for Steel Construction, many steel structure software packages now provide CIS/2 import and/or export capabilities.

The Industry Foundation Classes (IFC) is another industry standard related to building structure, which is developed by the International Alliance for Interoperability (IAI). The IFC provides data exchange and sharing capabilities for the building and construction industry [35]. The IFC is a data representation standard and file format for defining architectural and constructional CAD graphic data. The IFC uses text-based structures for storing the definitions of objects encountered in the building industry. While various domain models have been developed for the building industry, they are either too specific (e.g., CIS/2 for steel) or too general (IFC for all phases of a building project) to be useful for describing the specimens typically used in NEES experiments and simulations.

COSMOS geotechnical data model

The Consortium of Organizations For Strong- Motion Observation Systems (i.e., COSMOS, http://www.cosmos-eq.org/) recently completed a preliminary project on the dissemination of geotechnical data over the Internet [3]. The project had three major tasks, each of them assigned to a separate working group: (1) User Scenario; (2) Data Dictionary; and (3) Virtual Data Center. The Data Dictionary Work Group developed an extensible geotechnical data dictionary with XML formatting standards, and proposed some guidelines for assessing/evaluating data quality. The data dictionary included 35 tables, which were reduced to 12 tables in view of time constraints. The data dictionary, which defines in plain English all the variables used in the data model, is necessary to exchange successfully information among users with different backgrounds in engineering and seismology. The tables, intended for relational modeling, were translated into XML schemas, and data examples were compiled originating from various data providers, e.g., the California Department of Transportation, the California Geological Survey, and the United States Geological Survey. Applicable components of the XML schema have been adopted by at least one NEESgrid Field Test site (e.g., Brigham Young Univ.) in compiling their data model.

Oregon State Model
To our knowledge, the first comprehensive data model for NEES experiments was developed at the Oregon State University (ORST) and the Northwest Alliance for Computational Science and Engineering (NACSE). Based on the relational model and the concept of usability [36, 37], the ORST data model was initially designed for describing tsunami wave tank experiments [38]. Figure 6 shows a high-level E-R diagram for the ORST model.

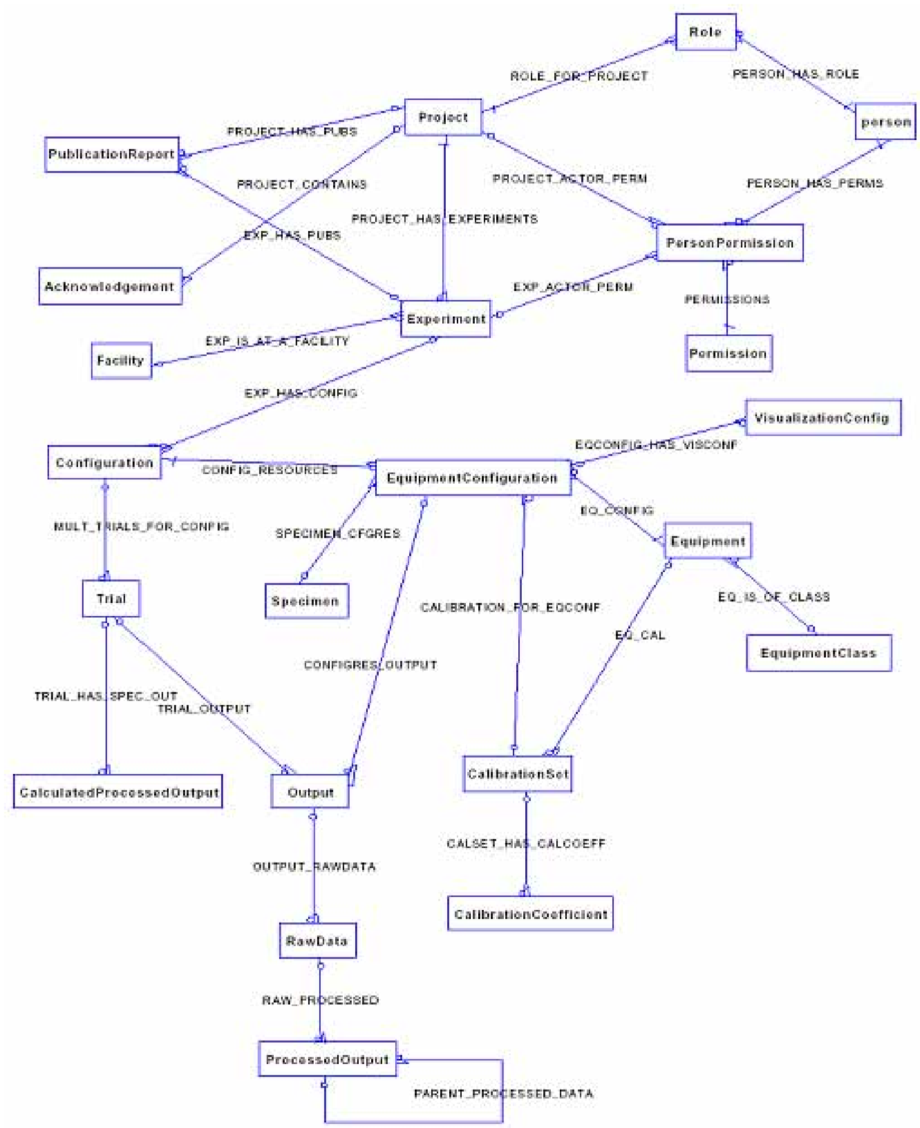

Figure 6. E-R Diagram of the ORST Model [38]

The ORST model contains relationships among projects, experiments, researchers, equipment, experimental results, etc. As shown in Figure 6, a project may have multiple experiments, an experiment may have multiple configurations, a configuration may have multiple trials, etc. The ORST model consists of a small number of entities, which makes the model simple and the number of tables manageable. The attributes of entities are assigned using a flexible scheme. For instance, the Equipment table in Figure 6 may contain any number of pieces of equipment used in an experiment (e.g., sensors) and does not assign a particular slot to each sensor. The ORST model was designed primarily for tsunami wave basin experiment and requires adaptations for other types of earthquake engineering experiments. It was however conceived to include new types of measurement instruments, and was successfully tested by independent researchers for structural engineering testing [39]. The simplicity and usability of the ORST model may appeal to researchers in earthquake engineering which had so far less experience in data modeling than other scientists. The ORST model provides many valuable lessons for developing more advanced NEES data models.

Centrifuge XML data model

A preliminary XML data model was developed for NEES centrifuge testing [40]. This data model was prompted by the surge in data volume generated by new sensor and data acquisition systems in centrifuge modeling. The centrifuge data model identifies all the objects that may participate to a centrifuge experiments and organizes them into a hierarchical structure. The main groups of objects are (1) Project identifiers; (2) Catalog of materials, objects, sensors and apparatus; (3) Sequence of model test events and measurements; (4) Sensor channel gain lists; (5) Image data; and (6) Control data files. The hierarchical structure was expressed using XML. The centrifuge data model points out directly the usefulness and versatility of object-oriented data modeling.

STRATEGY FOR DEVELOPING NEES DATA MODELS

The strategy for developing the NEES data models was conceived based on the existing methods for modeling data, and the lessons learned from other data modeling efforts in sciences and engineering. The NEES data/metadata committee decided to produce end-to-end solutions that integrate across site specifications database, project level model, domain specific data models, and common elements. Figure 7 summarizes the overall NEES data model.

As shown in Figure 7, the NEES data model represents the site specification database and project description based the ORST data model, which identifies well the different parties of the NEES collaboratory. The relationships among projects, experiments, trials, researchers, equipment, and experimental results of the ORST model can be retained and extended to incorporate other entities. The project-level information is being enhanced to incorporate some of the modeling details of the Ontology of Science.

The NEES data model has common elements, e.g., physical units, sensors, locations and time, which are represented using and extending many existing developments. For instance, it uses parts of SensorML which has detailed models on sensors and recorded data. It also uses the Berkeley CUREE/ Kajima model, which provides a comprehensive modeling on data types, quantities, and collectors. It also uses the Ontology of Science, which has effective means of representing location and time. Efforts are currently underway to consolidate the elements from the different models and to develop other essential elements to represent units and geometry.

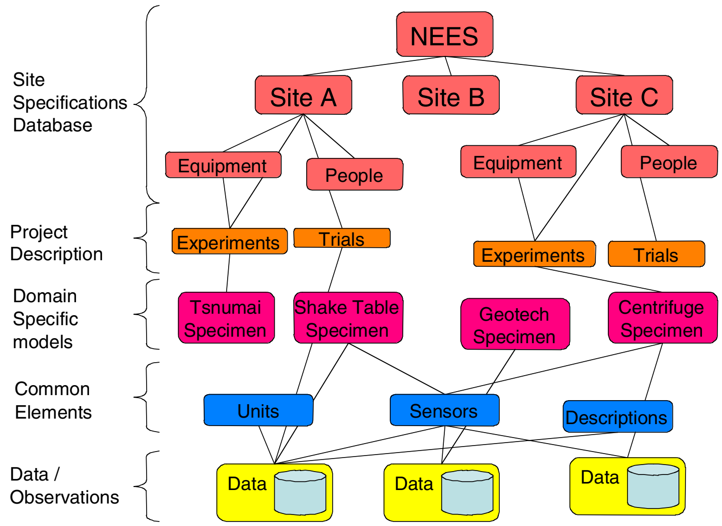

Figure 7. Overall Data Model for NEESgrid (Courtesy of C. Severance)

The NEES data models are developed using Protégé-2.0 as data modeling tool. Protégé-2.0 was selected because of its user-friendly GUI and its export capabilities to different formats, e.g., E-R diagrams, XML, and RDF. Figure 8 shows a recent version of the NEES data model using Protégé-2.0. Currently, Protégé-2.0 is only used to ingest relatively small data sets in earthquake engineering and to test the data models under development. Larger volumes of data will be ingested into the NEES data repository using the NEES metadata language NEESM [41]. NEESML presents many similarities to RDF; its syntax defines object types and objects, the values of their properties, and the relations between objects. The use of Protégé-2.0 to generate elements in RDF, which can then be exported as NEESML documents, appears to be a natural approach to develop NEES data models.

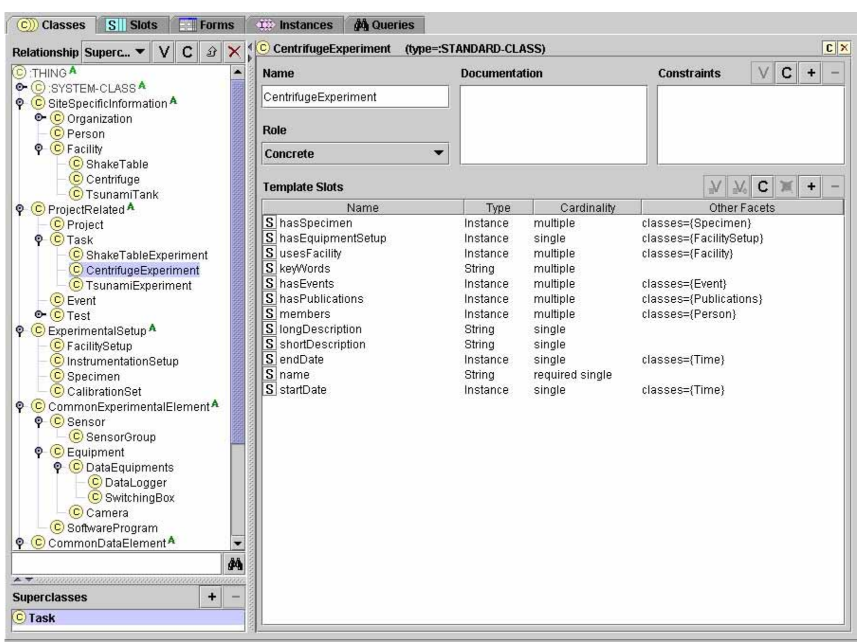

Figure 8. Preliminary version of NEES Data Model viewed using Protégé-2.0

The NEES data models are developed based on user surveys, usage scenarios and existing data sets from different fields of earthquake engineering. In parallel to the development of NEES data models, a NEES data dictionary is developed using Protégé-2.0. Data dictionaries define the terms in data models in plain English which can be understood across engineering disciplines. The NEES data dictionary is constructed based on the experiences from the COSMOS geotechnical data model project. At the present, the data dictionary is preliminary as it is developed by a small number of researchers. Its completion will require the involvement of the earthquake community at large including researchers and practitioners.

DISCUSSION

The present paper describes the ongoing effort related to data modeling, but does not cover other important data aspects such as electronic notebooks, and other ingestion tools. At present, the geometries of laboratory specimens are described using separate CAD files, which are integrated with drawings, notes, photos, narrative descriptions, electronic notes, and sensor data. In the future, it is likely that the laboratory specimens will have geometrical models more tightly linked to other types of data. Finally, it is important to point out that the needs and feedbacks from the NEES community/stakeholders are of paramount importance to the acceptance and usability of the NEES data models. The NEES data models will definitely evolve as they become tested and applied over time.

CONCLUSION

NEESgrid, the system integration of the NEES research collaboratory, uses recent advances in Information Technologies for modeling the data sets and experimental processes generated at the NEES testing sites. This document has reviewed several approaches and tools of data modeling, and has examined a few examples useful for developing data models for the NEES collaboratory. NEESgrid describes experimental results and processes using object-oriented data models expressed in NEESML, the NEES Metadata Language, which is a subset of RDF (Resource Descriptor Framework). At present, data models and data dictionaries are being developed for various fields of earthquake engineering using Protégé-2.0, based on user surveys, usage scenarios and existing data sets.

ACKNOWLEDGMENTS

The authors acknowledge the collaboration and contributions of the other members of the NEESgrid Data/Metadata task committee: Gokhan Pekcan, University of Nevada at Reno; Bill Spencer, University of Illinois at Urbana-Champaign; Charles Severance, University of Michigan; Andrei Reinhorn, State University of New York at Buffalo; and Lelli Van Den Einde, University of California at San Diego. The authors would like to thank Jim Eng of the University of Michigan and Joe Futrelle of University of Illinois at Urbana-Champaign for their contributions. The authors would also like to thank Patrick Laplace of the University of Nevada, Reno, and Steve Mahin, Bozidar Stojadinovic and Greg Fenves of the University of California, Berkeley for contributions to data issues in earthquake engineering experiments and simulations.

REFERENCES

1. ROSRINE. “Resolution Of Site Response Issues from the Northridge Earthquake.” Geophysical data sets; University of Southern California, Civil Engineering Department, Los Angeles California, 2001, http://geoinfo.usc.edu/rosrine/.
2. VELACS. “Centrifuge and laboratory data sets of the NSF collaborative project Verification of Liquefaction Analysis using Centrifuge Studies.” University of Southern California, Civil Engineering Department, Los Angeles, California, 1997, http://gees.usc.edu/velacs.

3. COSMOS GDM. "COSMOS Geotechnical Data Model, Version 0.5." University of Southern California, Civil Engineering Department, Los Angeles, California, 2003, http://geodata.usc.edu:1100/geotech_schemas/SatelliteSchema.html.
4. SAC, The SAC Steel project, http://www.sacsteel.org/, Structural Engineers Association of California (SEAOC), Applied Technology Council (ATC), Consortium of Universities for Research in Earthquake Engineering (CUREE).
5. CUREE-Caltech wood frame project, http://www.curee.org/projects/woodframe/, Consortium of Universities for Research in Earthquake Engineering (CUREE).
6. Bardet JP, Synolakis CE, Davies HL, Imamura F, Okal EA. "Landslide Tsunamis: Recent Findings and Research Directions." Pure and Applied Geophysics 2003; 160 (10-11).
7. Finholt T, Horn D, Birnholtz JP, Bae SJ, Motwani D., "NEESgrid User Requirement Document," Collaboratory for Research on Electronic Work, School of Information, University of Michigan, MI. http://www.neesgrid.org/documents/NEESgrid_User_Requirements_v3_0.pdf, 2003.
8. Date CJ. "An Introduction to database systems." 7th edition. Reading: Addison Wesley Longman, Inc., 2000.
9. Ullman J, Widom JA. "First Course in Database Systems." Prentice Hall, Inc., 1997.
10. Chen PPS. "The Entity Relationship Model – Toward a Unified View of Data." ACM Transactions on Database Systems 1976. 1(1):9-36.
11. Atkinson M, Bancilhon F, DeWitt D, Dittrich K, Maier D, Zdonik S. "The Object-Oriented Database System Manifesto." Proc. 1rst Int. Conference on Deductive and Object-Oriented Databases, Kyoto, Japan, 1989.
12. Rumbaugh J, Blaha M, Premerlani W, Eddy F, Lorensen W. "Object-oriented modeling and design." Englewood Cliffs: Prentice Hall, 1991.
13. Arlow J, Neustadt I. "UML and the Unified Process: Practical Object-Oriented Analysis and Design." Boston: Addison-Wesley Pub Co., 2001.
14. Humphreys BL, Lindberg DAB. "The UMLS project: making the conceptual connection between users and the information they need." Bulletin of the Medical Library Association, 1993; 81(2): 170.
15. Young MJ, "Step by Step XML." Microsoft Press, 2001.
16. McGuinness DL, Wright J. "Conceptual Modeling for Configuration: A Description Logic-based Approach." Artificial Intelligence for Engineering Design, Analysis, and Manufacturing - special issue on Configuration, 1998.
17. Benjamins R, Fensel D, Perez AG. "Knowledge Management through Ontologies." Proc. Of the 2nd International Conference on Practical Aspects of Knowledge Management, Basel, Switzerland, 1998.
18. Uschold M, Gruninger M. Ontologies: "Principles, Methods and Applications." Knowledge Engineering Review 1996; 11(2).
19. Musen MA. "Dimensions of knowledge sharing and reuse." Computers and Biomedical Research 1992; 25: 435-467.
20. Price C, Spackman K. "SNOMED clinical terms". BJHC&IM-British Journal of Healthcare Computing & Information Management 2000; 17(3): 27-31.
21. OWL Web Ontology Language, http://www.w3.org/TR/2004/REC-owl-ref-20040210/, World Wide Web Consortium, 2004
22. Lassila, Swick RR (eds.). "Resource Description Framework (RDF) Model and Syntax Specification." Recommendation, W3C, Feb. 1999. (http://www.w3.org/TR/1999/ REC-rdf-syntax-19990222.)
23. Brickley D, Guha RV. "Resource Description Framework (RDF) Schema Specification." Proposed Recommendation, World Wide Web Consortium 1999. http://www.w3.org/TR/PR-rdf-schema.
24. Duineveld AJ, Stoter R, Weiden MR, Kenepa B, Benjamins VR. "WonderTools? A comparative study of ontological engineering tools." International Journal of Human-Computer Studies 2000; 52(6): 1111-1133.
25. Protégé-2.0, "The Protégé Project." Stanford University 2004. http://protege.stanford.edu

26. Rothenfluh TR, Gennari J., Eriksson H, Puerta AR, Tu SW, Musen MA. "Reusable ontologies, knowledge-acquisition tools, and performance systems: PROTÉGÉ-II solutions to Sisyphus-2." International Journal of Human-Computer Studies 1996; 44: 303-332.
27. Gennari J, Musen MA, Fergerson RW, Grosso WE, Crubézy M, Eriksson H, Noy NF, Tu SW. "The Evolution of Protégé: An Environment for Knowledge-Based Systems Development." Technical Report SMI-2002-0943, Stanford Medical Informatics, Stanford University, 2002. http://www.smi.stanford.edu/pubs/SMI_Reports/SMI-2002-0943.pdf
28. Freitas, F. Ontology of science, http://derpi.tuwien.ac.at/~andrei/km/science_ont_freitas/index.html, and http://protege.stanford.edu/ontologies/ontologyOfScience/ontology_of_science.htm, 2001.
29. SensorML. Sensor Model Language, 2004. http://vast.uah.edu/SensorML/
30. Botts M. (ed.) "Sensor Model Language (SensorML) for In-situ and Remote Sensors." OpenGIS Interoperability Program Report, OGC 02-026, Version 0.7 Open GIS Consortium Inc, 2002. http://vast.uah.edu/SensorML/OGC-02-026_SensorML_0.07.doc.
31. ISO, "Product Data Representation and Exchange, Part 1: Overview and Fundamental Principles", No. 10303-1, International Organization for Standardization, 1994.
32. Stojadinovic B. "Framework for the integration and visualization of structural state data." University of California, Berkeley, CA 2003.
33. Eastman CM, "Building Product Models: Computer Environments, Supporting Design and Construction." CRC Press, 1999.
34. STEP, STandard for the Exchange of Product Data, http://www.eccnet.com/step/
35. IAI, "Industry Foundation Classes." Specification Volumes 1-4, International Alliance for Interoperability, Washington, D.C., 1997.
36. Pancake CM, Newsome M, and Hanus J, "'Split Personalities' for Scientific Databases: Targeting Database Middleware and Interfaces to Specific Audiences." Future Generation Computing Systems 1999; 16: 135-152.
37. Walsh KA, Pancake CM, Wright DJ, Haerer S, Hanus FJ, "Humane Interfaces to Improve the Usability of Data Clearinghouses." Proceedings of GIScience 2002; 333-345.
38. ORST data model. Oregon state University, Corvallis, OR 2004. http://nees.orst.edu/IT/data.model/
39. Erkman B, Hajjar JF, Daugherty DJ, Wan C. "Multi-Axial Subassemblage Testing System (MAST): Oregon State University Data and Metadata Model Applied to Quasi-Static Large Scale Testing," Report No. MAST-03-03, MAST Laboratory, University of Minnesota, Minneapolis, Minnesota, September, 2003, 35 pp.
40. Kutter B, Wilson D, Bardet JP, "Metadata Structure for Geotechnical Physical Models." Proceedings of the International Conference on Physical Modeling in Geotechnics, NewFoundland, 2002.
41. Futrelle J. "Overview of NEESML." Technical Report NEESgrid 2003-12, v1.0 National Center for Supercomputing Applications, Urbana-Champaign, IL, 2003. http://www.neesgrid.org/documents/NEESgrid_TR_2003-12.pdf.