Framework for Integration and Visualization of Structural State Data

Interim Project Report, February 20, 2002

The goals of this project proposed project are to conceptualize, develop and implement a framework for gathering, classification and integration of structural state data collected in and around a structure, and to enable effective visualization and fusion of such data to define the state of a structure.

Project Objectives

To achieve the stated goals, the following objectives are defined:

1. Define types of structural state data. Structural state data is differs by: source, scale, type, location and time. These data attributes are encapsulated using data type and meta-data concepts.
2. Define data structures for structural state data. An inheritance-based object-oriented data structure is established to generate meta-data to describe raw structural state data.
3. Define a data structure access paradigm. Access to a special and temporal database containing objectified structural state data is best accomplished using a visual access paradigm. A three-dimensional computer model of a structure is already a familiar visual paradigm used in structural analysis. This paradigm is extended to allow access to structural state data by its location and over time.
4. Verify using an experiment. Prototype implementations of structural state database and visual access tools will be verified and updated using an experiment.

Project Tasks and Schedule

The following tasks being executed in order to achieve project objectives stated above. As per June 2002 proposal revision, the tasks are defined in two groups to separate tasks for Project Year 1 and Project Year 2 and to insure planned project deliverables are ready for evaluation at the end of Project Year 1. The unfortunate consequence of this task separation is the inability to demonstrate the function of the structural state database on real data collected the CrossBow sensor and data acquisition system, as intended in the original proposal. Interaction with Kajima colleagues proved that such demonstration is very important. In response, Task 5 was started earlier than planned to provide at least some illustration of how to use the structure state database.

Project Year 1	Tasks	1. Review of data structures and access paradigms. 2. Define structural state data types and meta-data. 3. Design structural state data structures. 4. Implement structural state database prototype. 5. Implement prototype VR database access tool using VRML. 6. Link VRML access tool to structural state database. 7. Report on progress.
	Deliverables	1. Design of a meta-data for structural state data. 2. Prototype structural state database. 3. Prototype structural state VR access tool.

Project Year 2	Tasks	1. Build a model frame and install on a 1D shaking table located in UC Berkeley Structures Laboratory, 115 Davis Hall. 2. Instrument the model frame using CrossBow wireless sensors and Internet appliance moderate resolution video cameras. 3. Conduct long-term and short-term structural state monitoring experiments and store acquired data in the structural state database. 4. Refine implementation of structural state database. 5. Refine implementation of VR database access tool. 6. Report on progress.
	Deliverables	1. Improvements in structural state meta-data, database and VR access tool. 2. Example data collected on a physical frame model.

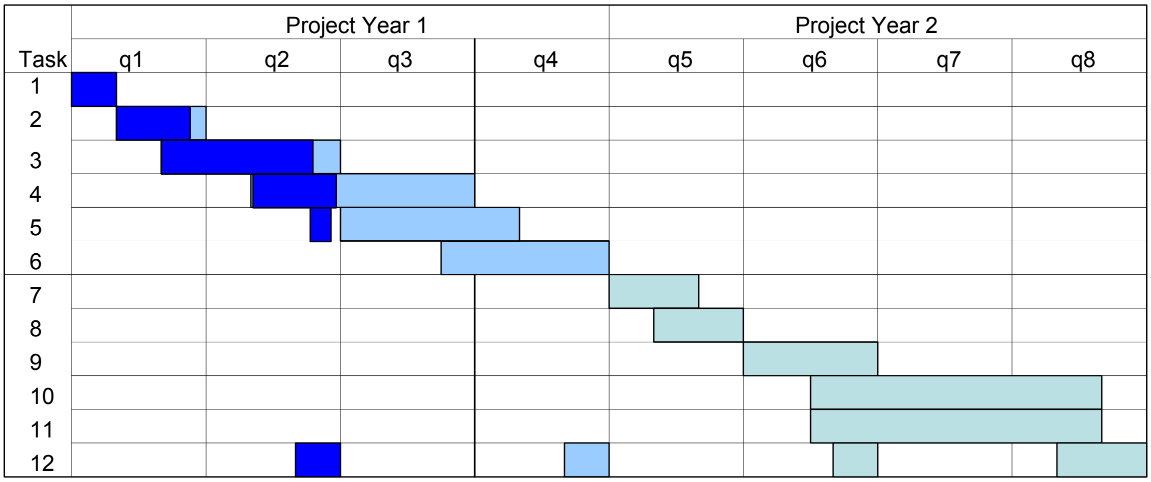

Figure 1. Project time-line. Task completion levels are shaded dark blue.

Project Status

At the end of the Second Quarter of Year 1 of the project, significant progress has been made on Tasks 1, 2 and 3, while Task 4 is ongoing, and preparations are made to start Task 5. Project information and current updates are posted on the project web site. http://www.ce.berkeley.edu/~boza/research/Curee-Kajima

Task 1: Review of data structures and access paradigms

The state of the structure is a characterization of structural integrity, structural safety and ability of the structure to perform its intended function. In engineering terms, such characterization is derived from the data collected in and around the structure before and after events that may cause a change of the state of the structure. Today, a multitude of sensors collect data relevant for the state of the structure. For example, acceleration records are collected during significant earthquake or wind events, while strain and temperature data are collected continuously. Video data, mostly used for security surveillance, can also be used to assess the state of a structure. Engineers inspect structures and collect data on material characteristics and physical appearance. Finally, numerical models are used to estimate the state of a structure.

To represent structural state data in a computer database, design requirements for such database were defined first in this Project Task. Structural state data is characterized by source, scale, time, location and data type. The data structure used to represent the state data must enable differentiation of these characteristics. Furthermore, the data structures must enable merging (fusion) of data collected at the same location or at correlated instants in time, regardless if the data is collected by physical sensors or by numerical simulation. Processing tools associated with the state database should enable simple forms of seismic risk assessment, such as comparison of material strength data collected over time, or identification of the change of the time-domain or frequency-domain response (such as period shift). The data structures should also enable interfacing to more complex damage assessment and mitigation software. Finally, the data structure must be general enough to accommodate new structures and types of data in the future.

A visual access paradigm is the best way for engineers to access the data and for clients to visualize the state of their structures. The clients need a visual tool to understand the state of the structures after a seismic event, to monitor the change of their structure over time, and to visualize the effect of a retrofit on the state of their structure. The engineer needs a more detailed look into the state of the structure, such as access to individual sensor readings and the ability to take collected data to numerical tools for structural modeling and assessment. Given the omnipresence of web browsers, it is quite important to enable access to the database using web-based tools. However, most of those tools offer 2-dimensional graphical representations. In case three-dimensional representation is required, a VRML representation is recommended.

Task 2: Data structures and meta-data

Structural engineering design requirements for the structural state database have been defined in Task 1. They defined the knowledge domain. Computer science design requirements to describe this knowledge domain were defined in this Project Task. First, a common vocabulary that enables sharing of information among people and software agents was defined. Such description (ontology) comprises concepts (classes), properties of each concept (slots) and ranges of property values (facets). This format is crucial for defining structural state concepts and relations among those concepts such that they can be represented in and used by machines. In addition, definition of ontology enables analysis and reuse of a domain of knowledge in the future.

Meta-data, or data about data, was used to describe the ontology of a structural state knowledge domain. Meta-data helps to define classes in the knowledge domain and arrange these classes in a taxonomic hierarchy with subclass/super-class relations and multiple inheritance. This structure is similar to the structure of objects used in object-oriented programming. Meta-data is also used to define characteristics of each class (slots) and specify allowable values for each slot (facets). Such description can be interpreted by machines as well as by humans.

An efficient was of representing the meta-data is to use an XML schema. The same approach is used by the NEES System Integrator to define the data structure for NEES data repository. This data structure was examined: it was found to be quite complex and too general for the structural state database developed in this project. The fairly well defined focus of the data involved in the description of the state of a structure makes it possible to simplify the NEES data structure and yet preserve the required functionality. Several ways of simplifying the NEES data structure were examined first. Then, more detailed data structure designs and meta-data definitions were implemented in Task 3 of this project.

Task 3: Design of structural state data structures

The data structure for a structural state database was designed using a software tool Protégé 2000. This tool enables the user to construct knowledge domain ontology and enter domain knowledge. It generates an XML schema using a graphical user interface and allows for quick changes and modification of the classes, slots and facets that define the ontology.

Several versions of the structural state database XML schema were made using Protégé 2000. Two versions are displayed on the project web site in HTML format. The February 10, 2003 version is currently used in Tasks 4 and 5 to implement and structural state database prototype and the visual interface. A graphic representation of this data structure is shown in Figure 2. More feedback from Kajima engineers is expected and will be used to modify the data structure before final implementation. A short description of the data structure is presented here, considering that the data structure is available for detailed examination at:
http://www.ce.berkeley.edu/~boza/research/Curee-Kajima/Schema-02-10-03/index.html

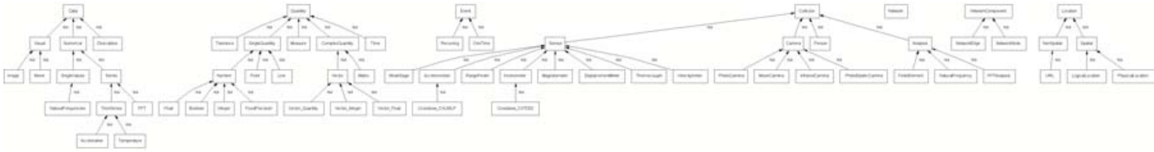

Figure 2. Data structure, February 10, 2003 version. Larger picture available of project web site.

The primary object in the data structure is the Data class. Data can be descriptive, numerical and visual. Data is collected during events that can be one-time or recurring. Furthermore, data is collected using a network of collectors. Network edges represent the connectivity among collectors, while the network nodes represent the collectors themselves. A collector may be a sensor, a person or a camera for collecting raw data, and an analysis tool (a black box whose internal workings are not of interest at this time) that transforms raw data into processed data. Calibration and collector state data is associated with every collector enabling a database user to verify the quality of the collected data. Location of collectors may be defined physically (by coordinates), and logically, by node numbers defined elsewhere, or in terms of a web resource, such as another database, an Internet Protocol address with live data, or a computer running an analytical simulation.

The collected data is characterized by the time-stamp when it was collected and by the format of what was collected. The time is described using an ISO 8601 time stamp format, the same format that was selected by NEES. The collected data is measured data represented using a number-unit pair. Such data may be scalar (single instance) or a vector of instances collected over time.

Task 4: Implementation of a structural state database prototype

A prototype of the structural state database prototype using the February 10, 2003 version of the XML schema was also implemented in Protégé 2000. The data stored in the database was generated during a shaking table test of a simple 4-story demonstration structure, shown in Figure 3. The data itself is available in an Excel file posted on the project web site. This data was treated as raw data collected from the sensors on the structure, even though in reality the data was pre-processed for the students taking the class.

Before the data was stored in the structural state database, it was re-formatted to the NEES ASCII format specified in NEESgrid-2003-03 report "Protocol specification for NSDS, Driver and DAQ". This report is also posted on the project web site. It is expected that data generated by collectors will always be formatted in a standard format before being stored in the structural state database. At this time, the NEES ASCII format was chosen. Compatibility with NEES will be maintained as the NEES formats change.

Once data was converted to a standard format, it was placed into Protégé 2000. For example, a time history recorded by an accelerometer located at the top floor of the demonstration structure is stored at: http://www.ce.berkeley.edu/~boza/research/Curee-Kajima/Schema-02-10-03/dataProject1_00263.html using two arrays, one containing the acceleration values, the other containing the ISO 8601 time-stamps for each acceleration value. Also available in the database are FFT spectra of the white noise response that can be used to extract the periods of the structure by using a signal processing tool available as a "black box" in the Collector-Analysis-FFTAnalysis slot of the data structure. Other "black box" analysis collectors can be inserted into the database structure in the same manner. Such collectors may be building-specific or generic, but will use the data stored in the structure state database as input and will output their results into the structure state database.

Figure 3. Class demonstration of a shaking table test.

Two problems with this implementation were encountered. First, the entire XML schema and the data were kept in the same file. This file quickly became large and hard to maintain and transfer. Second, the size of the file exceeded the size of the memory heap allocated for some of Protégé 2000 arrays. The Protégé 2000 software was recompiled with somewhat larger memory size structure only to provide a temporary solution. The long-term solution for the structural state database is an SQL database implementation. The XML schema developed in Protégé 2000 will is to develop the structure of SQL tables. At this time, installation of an SQL database served is completed and conversion of the Protégé 2000 XML schema is ongoing.

After the database structure is implemented on an SQL server, work in this Project Task will focus on developing automated data re-formatting, data input and data output drivers. These drivers will enable other software to input and extract data from the structure state database. Once the drivers are completed, simple graphing routines will be used to graph the data extracted from the database. These routines will be a part of the database access and interface tool.

Task 5: Database access and interface tool

Following the request of Kajima colleagues to illustrate the application of the structural state database in their interaction with Kajima clients, work on this Project Task was started earlier than originally planned. Illustrations of how to use the structural state database are made by defining use scenarios. A use scenario describes how a potential user will interact with the structure state database.

At this time, only one user scenario was developed to illustrate the interaction between Kajima engineers and Kajima clients using the structural state database. This interaction is will take place using a standard Internet browser, such as Internet Explorer or Netscape. Sketches of the browser screens are available on the project web site. An interaction scenario is presented in this report. Development of other use scenarios is planned after further consultation with Kajima engineers.

A structure designed, built and instrumented by Kajima will have a computer server designed to host the structural state database. Collectors (sensors, cameras) installed in the building will automatically put data into the state database. The server machine will also be used to serve web pages to the Kajima client (owner of the building) to present the state of the structure.

An opening web page will show the name of the building, a street map showing the orientation of the building plan and a picture of the building. A separate frame will show the current state of the building, represented by damage index indicator dials. In this example, dials have green-yellow-red divisions to give Kajima clients an immediate sense of the state of the structure. The type of damage indicator displayed may be a measure of the change of the fundamental period of the building, a recent accelerometer recording, a recently measured vibration period, or other damage index measures derived from the data stored in the database. Such damage index measures are building-specific and thus would have to be designed by Kajima engineers and build into the structural state database.

The opening web page alone may be sufficient to the Kajima client to quickly learn about the state of the structure, in the same way that a driver can quickly learn about the state of his or her automobile just by looking at the dashboard. The third frame of the opening web page will have buttons allowing access to architectural and structural drawings of the building, finite element or other numerical models of the structure, and the data in the structure state database. It may be best that a Kajima engineer help the Kajima client while viewing such more complex information.

The data in the structure state database may be accessed in two ways: though the event calendar web page and through the collector location web page. The event calendar is designed to look like an appointment organizer, with events represented as appointments that occurred on a given date and time. This interface paradigm will be used to define the time aspect of the structural state data and to locate the time-stamp of the event for further viewing and analysis.

The collector location page will have a wire-frame drawing of the structure in its center. This may be a set of AutoCad-like 2-dimensional drawings or a 3-dimensional virtual reality model of the structure. Both graphical representations will allow the user to zoom in and out. Collectors will be located on the structure using the location data provided in the database. Graphically, they will be represented using different icons, according to collector type. A switch panel on this web page will allow the user to remove or insert a group of collectors on the drawing and thereby control the density of displayed information. This way, the collector location page will display the spatial aspect of the structure state data.

One collector location page will be generated for one data collection event. The event time-stamp will be chosen using the event calendar. Thus, if a Kajima engineer wants to compare the state of the structure after two different events, he or she would open two collector location web pages.

Left-click on the collector icon in a collection location web page will bring up a graphical representation of the collected data. For example, left-click on an accelerometer will bring up an acceleration time-history recorded during the chosen event. Another example is a left-click on the camera icon that would bring up a video recorded during an event. Right-click on the collector icon will open the web page associated with that collector. In this web page, data related to collector type, status and calibration will be available. In addition, data collected by this collector in all recorded event may be accessed and downloaded. Choosing multiple collectors at once (shift-left-click) will bring up an appropriate simultaneous display of the sensor data. For example, choosing two collectors in the collector location page for the same event would plot their data in an x-y plot. Another example is a choice of identical collectors in collector location pages for different events: in this case, two signals will be plotted on the same time-history plot for direct comparison.

Remaining Year 1 tasks and Year 2 tasks

Work during the remaining time in Year 1 will focus on three principal issues. First, current database XML schema will be implemented on an SQL server, together with access driver routines. Second, user scenarios will be developed an implemented such that the use of the structure state database can be demonstrated to Kajima engineers. Artificially created data will be used in these demonstrations. Third, improvements in the data structure and XML schema will be made as needed to improve the performance of the database.

The proposed work-plan for Year 2 anticipates the purchase and installation of a CrossBow accelerometer and data acquisition system. Such system will enable generation and collection of data using a structure on a shaking table simulating an actual building situation. Structure state database application programmer interface (API) will be developed to enable communication between the database and the data acquisition system and between the database and other software, such as an application selected by Kajima engineers. Such working system will enable further improvement of the database structure and refinement of use scenarios and interface tools: an example of user interface refinement may include a comparison of a structure before and after a proposed seismic retrofit.

Significance for Seismic Risk Assessment and Mitigation

The structure state database enables collection of diverse data collected in an around the structure in a one place using a common format. Together with database access and interface tools, the structural state database will let the building owner observe and monitor the state of the structure, making quick seismic risk assessment possible. Furthermore, ability to access the collected data and apply more sophisticated assessment, design and analysis tools will enable Kajima engineers to inform the building owner about possible measure to improve the structure and demonstrate the effect of such measures on mitigating the seismic risk.