Framework for Integration and Visualization of Structural State Data

This document is written to further explain the CUREE-Kajima project "Framework for Integration and Visualization of Structural State Data".

Background

My project is based on the fact that a significant amount of data is collected in and around a structure. Such data is collected for very diverse needs, such as: environment management in buildings, network status in Ethernet networks, building security, quality control of a manufacturing process, wind response control, and earthquake response monitoring. Structural engineers and architects are typically aware only of the last two data sources. Yet, we can potentially use all of the available data sources, and possibly install new ones that are missing, to gain an excellent view of the state of the structure.

Bridge Monitoring Systems

A good example for a state monitoring system is a bridge management system. Such system for collecting data and monitoring bridges has been developed by FHWA in the past decade (http://www.fhwa.dot.gov/infrastructure/asstmgmt/manag.htm). The software component of this system, called Points, is a data collection, management and presentation tool (http://aashtoware.camsys.com/html/PontisReport.html). The result of consistent application of such systems is a US National Bridge Inventory study (http://www.nationalbridgeinventory.com/), an amazing geographic database of bridge condition.

A different type of bridge monitoring system, one with real-time properties, is the structural monitoring system implemented on modern long-span bridges, such as the Akashi Kaiko Bridge in Japan or the Alamillo Bridge in Span. Similar systems are installed in most modern long-span bridges. Existing bridges are instrumented, too. A good example is the CSMIP effort to instrument the Golden Gate Bridge (http://www.consrv.ca.gov/cgs/smip/about.htm).

There are many ways of collecting data relevant to structural performance. Kinemetrics is a well-know company that produces both hardware and software to accomplish such tasks (http://www.kinemetrics.com/home.html). Wireless sensors (http://robotics.eecs.berkeley.edu/~pister/SmartDust/).
and systems for their integration (http://www.citris.berkeley.edu/technology/sensors/tinyos.html) are being developed at UC Berkeley's CITRIS (http://www.citris.berkeley.edu/).

Data Fusion

Collecting all available data may not be a problem. However, understanding a potentially very large amount of data in terms of what it means for a structure is a problem.

Similar problems exist in many other engineering and science fields. The general approach to solving them is data fusion. Data fusion is the formal framework that expresses the means and tools for the alliance of data originating from different sources. It aims at obtaining information of greater quality, the exact meaning of which will depend on the application (L. Wald, TGRS vol 37 no. 3, p. 1191). Another definition comes from NOAA: data fusion is the seamless integration of data from disparate sources. The data have been integrated across data collection "platforms" and geographic boundaries, and blended thematically, so that the differences in resolution and coverage, treatment of a theme, character and artifacts of data collection methods are eliminated. At present, this is a desirable but unattainable goal. Data fusion is the opposite of data fission. Data fission could be considered the result of developing separate data sets from a single source of data, such as the NOAA Polar Orbiting Environmental Satellites. The data are "separated at birth" for storage in different locations. Data fusion would be the process of re-joining, or integrating, these data. Ideally, the ultimate process would re-evaluate the original separation of the data, and revise the processing technique to keep the data integrated, or to facilitate the re-integration of the data.

A web search for “data fusion” will reveal diverse applications, both civilian and military (http://www.data-fusion.org). A good overview of data fusion in the field of Nondestructive Testing is available in a book “NDT Data Fusion” by X. E. Gros, Wiley, 1997.

Research Opportunity

Today, buildings are well instrumented structures. Various data is continually collected, yet structural engineers and architects have limited access, and therefore, limited knowledge of the state of the building structure. There is an opportunity to conceptualize, develop and implement building management systems similar to bridge management systems. The fundamental characteristic of such systems should be their ability to provide structural state data to structural engineers and architects.

A framework for data management is at the core of a building management system. Such framework is necessary because tools for data gathering and classification on one end, and tools for data fusion, analysis and representation on the other end, must interface and cooperate in a meaningful manner.

Project Goals

The goals of this project are:

1. To conceptualize, develop and implement a framework for gathering, classification and integration of structural state data collected in and around a structure; and
2. To enable effective visualization and fusion of such data to define the state of a structure.

Project Tasks

This project comprises the following projects tasks. The tasks addressed in this document are selected on the basis of the fact that funding for only the First Year of the project has been granted. They are listed in the "Projects Tasks and Schedule" section of my proposal. My proposal for this project contains an outline of my plans to continue the project in the Second Year. However, at this point in time, such continuation is not certain.

Data Types

Data types comprise the information about the data that can be collected in a building structure. There are a number of attributes that can be associated with such data:

1. **Source**. Data can be collected at a single point or at for a number of points that represent a filed. A single-point data source is a strain gauge or a temperature sensor. A field data source is a camera. Typical video cameras acquire visual information; however, infrared cameras acquire temperature-field data. Another source of field data may be computer software designed to generate field parameters based on single-point sensor data. An example of such computer-generated field is a temperature filed in a room computed on the basis of a small number of temperature and air-flow measurements using a computer model.
2. **Scale**. The scale at which data is collected defines the scope addressed by the data, and its use in structural state evaluation. Local-scale refers to data collected at centimeter and sub-centimeter scale. Examples of such data are: strains measured by a strain gage, temperatures measured by a temperature sensor, non-destructive test data, and material property data. Intermediate-scale refers to data collected at a meter order-of-magnitude. Examples of such data are rotation of connections and deformation of structural elements, such as braces or girders, field data related to temperature and air-flow, and data collected during the construction of the structure. Global-scale refers to data collected at 10-meter and larger order-of-magnitude. Examples of such data are inter-story and structure drift data, environment temperature data, and structural design documents.
3. **Type**. Data type describes the form data is in. Discriminating categories in this case may be related to types of measurement (analog vs. digital, continuous vs. discrete), and to modes of computer data representation (single integer, array of doubles, matrix).

4. **Location**. The information about where the data was taken is essential. This information locates the data source in the space of the structure and enables spatial correlation among data. Location data is, also, an essential ingredient in the data framework structure.
5. **Time**. Information about when data was taken in also essential. This information locates the data in time and enables synchronization of data in time. This feature is, also, essential for the data framework structure.

Data Structure

Metadata is information about the data. It describes the properties of the data, therefore it is quite open-ended. Metadata may contain a description of the source, scale, type, location and time attributes of the data. However, the most important use of metadata is the definition of the structure of the data.

A good way to define the structure of data is to use an inheritance-based model. This model defines the schema by which data can be classified. The structure of such classes is similar to classes found in object-oriented programming languages. At the root of the class structure are general identifier classes, with version and document attributes. These classes are general. Specific classes directly related to a structural state database, such as drift or acceleration measurement logs, are the leaves of a tree-like class inheritance structure.

An example of metadata structure created to describe a test of a composite structure is attached. This example has been developed within the NEES Project (http://www.nees.org). NEES, the George E. Brown Jr. Network for Earthquake Engineering Simulation, is an NSF Major Research Equipment project. The goal of this project is to enable collaboration and explicit integration within its programs of experimentation, theory formulation and validation, data curation, model-based simulation, high performance computing and education. The outcome of the project is expected to be a substantial acceleration of development of technically sound and cost effective approaches to earthquake loss reduction.

My plan is to use the NEES prototype metadata structure as a starting point in developing the metadata structure for this project. I plan to use standard Resource Description Framework (http://www.w3.org/RDF/) schema, and a tool called Protégé (http://protege.stanford.edu/index.html) to create the metadata structure. Then, I plan to implement such a structure using XML. At this stage, I do not plan to develop a relational database. Instead, I plan to use the hierarchy of the file storage system.

The NEES metadata structure allows for localization of data in three-dimensional space of a specimen. This data is useful to describe structural state data. However, a particularly important feature of the metadata structure is the temporal information. Such information is essential for time-synchronization of the structural state data, giving it the fourth dimension.

Interface

Access to structural state data is accomplished through a user interface and a application programmer's interface. The interface is defined first by choosing an access paradigm.

Considering that the primary users of a structural state data are structural engineers and architects, people used to geometric representation of data, a using a visual access paradigm is suitable for a user interface. Such paradigm is based on using visual queues for data searching, selection and representation. In the case of a structural state database, a two-dimensional drawing or a three-dimensional model of a structure is an appropriate interface format. In either format, sources of data are geometrically localized at the locations that correspond to physical locations of data sources in the structure. Access to data is enabled by equipping the drawing or the model with buttons that activate retrieval of the appropriate data and its presentation in the appropriate form.

For example, a drawing of the structure may contain an accelerometer positioned at the roof. Clicking on the accelerometer button would bring up the graphs presenting the data recorded in the database. In the same example, a drawing of a camera may represent an actual video camera used in the structure for security purposes. Video recorded from the camera may be accessed through appropriate viewing software. Time synchronization of recordings would enable the viewer to simultaneously view the video and the acceleration data recorded during an earthquake. This is a form of data fusion.

At this stage, I am considering two interface designs. One design is based on two-dimensional (plan-like) images enhanced with buttons to enable access to recorded data. Such interface can be implemented in standard browsers, while data retrieval from a server requires no more than generating a graph on a pop-up web-page. While the server side of this process may be somewhat computationally involved, the client would be a very light one. In fact, similar applications, such as retrieval of stock price information, are in general use. At this time, I am considering an implementation using Microsoft's .NET architecture (http://www.microsoft.com/net/basics/whatis.asp).

Another design is based on a three-dimensional representation of the structure. I am planning to use the Virtual Reality Modeling Language (VRML) to implement such representations. Virtual Reality Modeling Language is an international standard for describing 3-D shapes and scenery on the World Wide Web (http://www.sdsc.edu/~nadeau/Courses/Siggraph98vrml/vrml97/vrml97.htm). VRML's technology has very broad applicability, including web-based entertainment, distributed visualization, 3-D user interfaces to remote web resources, 3-D collaborative environments, interactive simulations for education, virtual museums, virtual retail spaces, and more. Building a Virtual Reality Model of the structure would enable the user to examine the structural state data in a virtual environment, in effect "fly through" the structure and look at the data. Software already exists to convert computer-aided drawings compliant with the CIS2 standard to VRML (http://cic.nist.gov/vrml/cis2.html). VRML models can be displayed on a wide variety of platforms, from regular browsers to the CAVE Virtual Reality System (http://www.evl.uic.edu/pape/CAVE/). Such three-dimensional representation is as exciting as it may be distracting: professional users may prefer a simple and straight-forward two-dimensional interface.

The application programmer's interface (API) enables other software to access the structural state data. Definition of the metadata is crucial for designing the functions to store and access structural state data from a repository. At this time, I plan to develop a prototype API sufficient for basic data storage and retrieval from a typical server.

Topics for Discussion

During our October 7 and 8 meetings, I would like to discuss the principal tasks of my proposal with Kajima researchers. In particular, I would like to address the following topics:

1. Data types. I am interested in what kind of data is regularly collected in buildings Kajima designs and builds. For example, what types of sensors and methods for data acquisition and storage are used? Feedback on source, scale, type, location and time of data is appreciated.
2. Data structure. If there is a data structure that already exists at Kajima, I would like to see an example. Also, please examine the example I provided (attached files) to see if this system can be extended from a level of a single experiment to the level of collecting and presenting structural state data.
3. Interface. If there is a building management system used at Kajima, I would appreciate seeing one in action. Feedback on the form of the user interface and a discussion on how useful a Virtual Reality interface would be are needed.

I am looking forward to more discussion regarding our meeting and an excellent exchange of ideas in Tokyo.