USD - a Database Management System for Scientific Research

Rowland R Johnson, Mandy Goldner, Mitch Lee, Keith McKay, Robert Shectman and John Woodruff

Lawrence Livermore National Laboratory
Livermore, California 94550

USD (Unstructured Scientific Data) is a database system developed at Lawrence Livermore National Laboratory (LLNL) that provides database capabilities required when doing scientific research. USD is implemented in a version of EMACS Lisp that has been extended to include a relational database management system and graphical user interface primitives. It is currently being used by several different scientific research efforts at LLNL.

Current data models require that data be manipulated in terms of a predefined structure. Many scientific database applications have structured data, but, also have unstructured data that is not easily organized. The need for unstructured data arises in applications where the organization is unknown or continually changing.The existence of this type of data appears to be a driving force behind the development of object oriented data models and the extended relational data model. In essence, the strategy employed by these models is make complex structure easier to deal with. However, structure must be pre-specified and adhered to. That is, prior to data entry, the data organization must be determined and all data must be expressed in terms of that structure. From this perspective, these data models are not much different from the relational data model (or from the network and hierarchical data models). The crucial aspect of this work is in accepting the unstructured view of this data.

Another aspect of scientific research that USD addresses is the widespread use of pre-existing data analysis tools. Both in-house as well as commercial products are in use. In general, these tools are used in both batch and interactive modes. USD provides a mechanism whereby it is easy to construct an interface for a new tool without having to modify the tool itself. The tool can then be run as a foreign process to USD and be operated in either batch or interactive modes.

The video presentation shows three examples of USD being used to analyze seismic data from an underground nuclear explosions.

The first example introduces the basics required to use USD. USD employs semantic networks as a user data model. A semantic network is a labelled directed graph wherein nodes represent objects and links represent the relationships between those objects. There exists an obvious, well defined mapping from a relational representation to a semantic network representation. Semantic networks can also be used to represent unstructured data. Data retrieval is illustrated by specifying a semantic network pattern (i.e. a semantic network with labelled links but no objects at the nodes) and directing USD to *match* that pattern. The resulting data is displayed as a sequence of semantic networks. Data is also displayed in tabular form. Finally, unstructured data is appended through via the graphical user interface.

The second example illustrates the capability to use a foreign process in order to analyze data. First, a pattern is constructed by merging two existing patterns and then modifying it to match just the data required. A procedure pattern is a representation of a procedure wherein a single node represents the procedure and its incoming and outgoing links represent the input and output arguments, respectively. Data patterns are white and procedure patterns are colored (cyan for a non-aggregate procedure and green for an aggregate procedure). A procedure pattern that plots one vector against another is selected and merged with the data pattern. After directing USD to match the data pattern the procedure is then invoked. In order to plot different data the procedure pattern is re-connected to a different part of the data pattern and then re-run. Finally, direct interaction with the foreign process is illustrated.

The third example illustrates some of the more complex capabilities of USD. A data pattern is constructed by taking two existing data patterns and moving a node from one pattern over a node from the other pattern. The merged pattern corresponds to relational query with a join clause. Restrictions on the matched data are achieved by placing predicates on the nodes of the resulting pattern. A new procedure pattern is created using the graphical user interface to create the necessary node and links. After specifying the procedure, the links are then connected to the data pattern. Unlike the previous example, this procedure produces output that needs to be saved in the database. This is achieved by adding a sub-data pattern that represents the structure of the data that will be produced by the procedure. Thus, this same data pattern can be used to subsequently match this new data. The procedure is run iteratively, once for each instance matching the data pattern. The results are considered to be unstructured since it is unknown by USD how many new data with the same structure will be created. However, upon completion of the iterative invocation of the procedure USD detects the existing structure and offers to convert the newly created unstructured data to structured data.