Implementation of PepcDB Reporting at CESG: More Trials and Fewer Tribulations

Craig A. Bingman, Xiaokang Pan, Gary Wesenberg, and George N. Phillips, Jr.

University of Wisconsin-Madison, Department of Biochemistry, 433 Babcock Drive, Madison, Wisconsin, USA 53706-1549, http://www.uwstructuralgenomics.org

Abstract
The Protein Expression and Purification Database (PepcDB) was created to capture data from the Protein Structure Initiative’s Centers. The idea is that such data, including positive and negative results from cloning, expression, purification, and crystallization trials would advance the field and encourage complementary efforts among scientists. PepcDB extends the content of the target selection database (TargetDB) by including status history, stop conditions, test protocols, experimental details, and contact information. The specification of data fields is fairly mature and stable, consisting of an XML schema which is documented by the Protein Data Bank (http://pepcdbpdb.org/).

The Center for Eukaryotic Structural Genomics (CESG) has been proactive in developing experimental tracking data, primarily with its Sesame laboratory information management system. Sesame gathers data, allowing a complete trace from initial selection to final publication. Standard reports from Sesame are then used to prepare a weekly PepcDB update file. Our most recent report contained experimental details for each distinct target on a total of 7370 CESG targets.

We have found that the concept of a directed graph provides a suitable framework for both visualizing our workflow through multiple trials on a given target, and preparing XML reports for PepcDB. In particular, work on each target can be described as a finite directed acyclic graph. The sources are selection actions, and the sinks are the most advanced pipeline actions for each trial. An algorithm has been devised that accurately identifies sinks and correctly traces each experimental trial back to its source. Significantly, our current approach does not require that targets flow hierarchically toward more “advanced” pipeline stages, supporting recycling of targets into multiple destination vectors at the cloning stage, lateral transfers of samples between NMR and cryocrystallography, and “on the fly” creation of novel experimental paths. CESG has leveraged the substantial body of work devoted to rendering paths. CESG has leveraged the substantial body of work devoted to rendering

Core Concepts

It seems important to enforce directed associations between primary data elements in a database for structural genomics. This is most simply implemented as hierarchical relationships, with one parent having one or more children. Such relationships can be implemented in many types of databases, including familiar relational databases. Additional associations can be present, but it is extremely helpful to have the primary parent-child relationships made explicit and unambiguous by end users.

In the implementation of Sesame at CESG, most of these hierarchical relationships exist between individual elements (indicates, wells, targets) in workgroups, and this design is used for the same targets in a parent workgroup. Most often, database workgroups map to physical 96-well plates, but the abstraction of a workgroup of elements having independent parentage can be applied to other units of work. This model supports refactoring, merging, and bifurcation of individual targets between workgroups.

The existence of hierarchical relationships between elements of workgroups leads naturally to expressing the data as a tree or graph. While XML can describe many types of graphs (cyclic elements are possible in XML documents via the use of DREFT), we prefer to conceptualize our XML reports to PepcDB as finite, directed, acyclic graphs. As such, it is much easier to understand how the data is related, and to see which targets (or sub-targets) have been reported, or are still undergoing work.

It seems important to enforce directed associations between primary data elements in a database for structural genomics. This is most simply implemented as hierarchical relationships, with one parent having one or more children. Such relationships can be implemented in many types of databases, including familiar relational databases. Additional associations can be present, but it is extremely helpful to have the primary parent-child relationships made explicit and unambiguous by end users.

In the implementation of Sesame at CESG, most of these hierarchical relationships exist between individual elements (indicates, wells, targets) in workgroups, and this design is used for the same targets in a parent workgroup. Most often, database workgroups map to physical 96-well plates, but the abstraction of a workgroup of elements having independent parentage can be applied to other units of work. This model supports refactoring, merging, and bifurcation of individual targets between workgroups.

The existence of hierarchical relationships between elements of workgroups leads naturally to expressing the data as a tree or graph. While XML can describe many types of graphs (cyclic elements are possible in XML documents via the use of DREFT), we prefer to conceptualize our XML reports to PepcDB as finite, directed, acyclic graphs. As such, it is much easier to understand how the data is related, and to see which targets (or sub-targets) have been reported, or are still undergoing work.

Implementation of Sesame at CESG, most of these hierarchical relationships exist between individual elements (indicates, wells, targets) in workgroups, and this design is used for the same targets in a parent workgroup. Most often, database workgroups map to physical 96-well plates, but the abstraction of a workgroup of elements having independent parentage can be applied to other units of work. This model supports refactoring, merging, and bifurcation of individual targets between workgroups.

The existence of hierarchical relationships between elements of workgroups leads naturally to expressing the data as a tree or graph. While XML can describe many types of graphs (cyclic elements are possible in XML documents via the use of DREFT), we prefer to conceptualize our XML reports to PepcDB as finite, directed, acyclic graphs. As such, it is much easier to understand how the data is related, and to see which targets (or sub-targets) have been reported, or are still undergoing work.