Rapid integration of genomic data from multiple sources using XMLGUS (An Improved Interface to the Genomics Unified Schema)

Yi Jia, Josef Jurek, Terry Clark, and Kamen Nikolov, Information Telecommunication Technology Center/Electrical Engineering and Computer Science Department, University of Kansas, Lawrence, KS 66045, Department of Molecular Genetics and Cell Biolology,The University of Chicago, Chicago, IL 60637. 2006 Annual Kansas City Area Life Sciences Research Day.

The object-oriented database Genomics Unified Schema, GUS, provides an extensive genomics oriented database schema with an object-oriented interface and centralized software control for a growing end-user community. GUS satisfies critical genomic database needs for numerous projects; yet, obstacles to facile data integration can interfere with its effectiveness. Software development involving relational databases for genomics projects is difficult. We believe that lowering this barrier is crucial for enabling scientific discovery. At present, to implement a novel query with new data and relationships requires significant expert time. One of our goals with XMLGUS is to make genomics database interaction easier if not routine. With GUS, the object layer can get bogged down with source-specific input modules, and new schema and data formats are tedious to incorporate. Site maintenance increases accordingly. We will present our system which addresses these problems. Input data are self-describing with respect to the schema using an XML format, GUS XML. The interpreter, XMLGUS, eliminates the error-prone task of writing GUS compliant data plugins. Instead, the common format is automatically handled by the extended GUS object layer. This system unifies the GUS data interface, provides the basic mechanism for extending the GUS schema automatically for new data types, and makes GUS amenable for interoperation. In this poster we discuss database development and application issues, along with details of the implementation. We also present motivating examples of projects using XMLGUS. We will also present performance data. Applications include a high-throughput genome sequencing and annotation project using the canonical GUS schema.