Design of a data model for developing laboratory information management and analysis systems for protein production

ARTICLE *in* PROTEINS STRUCTURE FUNCTION AND BIOINFORMATICS · FEBRUARY 2004

Impact Factor: 2.63 · DOI: 10.1002/prot.20303 · Source: PubMed

Anne Pajon
15 PUBLICATIONS **1,184** CITATIONS
SEE PROFILE

Cathy Lawson
Rutgers, The State University of New Jersey
47 PUBLICATIONS **2,949** CITATIONS
SEE PROFILE

John Westbrook
Rutgers, The State University of New Jersey
126 PUBLICATIONS **33,670** CITATIONS
SEE PROFILE

Ernest Laue
University of Cambridge
125 PUBLICATIONS **7,570** CITATIONS
SEE PROFILE

Design of a Data Model for Developing Laboratory Information Management and Analysis Systems for Protein Production

Anne Pajon,[1] John Ionides,[1] Jon Diprose,[2] Joël Fillon,[1] Rasmus Fogh,[3] Alun W. Ashton,[4] Helen Berman,[5] Wayne Boucher,[3] Miroslaw Cygler,[6] Emeline Deleury,[7] Robert Esnouf,[2] Joël Janin,[8] Rosalind Kim,[9] Isabelle Krimm,[10] Catherine L. Lawson,[5] Eric Oeuillet,[8] Anne Poupon,[8] Stéphane Raymond,[6] Tim Stevens,[3] Herman van Tilbeurgh,[8] John Westbrook,[5] Peter Wood,[4] Eldon Ulrich,[11] Wim Vranken,[1] Li Xueli,[6] Ernest Laue,[3] David I. Stuart,[2] and Kim Henrick[1]*

[1]EMBL Outstation, European Bioinformatics Institute, Wellcome Trust Genome Campus, Hinxton, Cambridge, United Kingdom
[2]Oxford Protein Production Facility, Wellcome Trust Centre for Human Genetics, Headington, Oxford, United Kingdom
[3]Department of Biochemistry, University of Cambridge, Cambridge, United Kingdom
[4]Daresbury Laboratory, Daresbury, Warrington, United Kingdom
[5]Department of Chemistry and Chemical Biology, Rutgers, The State University of New Jersey, Piscataway, New Jersey
[6]Biotechnology Research Institute, NRC, Montréal, Québec, Canada
[7]Plateforme Transcriptome CNRS/INRA de Sophia Antipolis, Sophia Antipolis, France
[8]LEBS, Equipe Génomique Structurale, Gif-sur-Yvette, France
[9]Lawrence Berkeley National Laboratory, Berkeley, California
[10]Laboratoire de RMN Biomoléculaire, Université Claude Bernard, Lyon, France
[11]Department of Biochemistry, University Wisconsin Madison, Madison, Wisconsin

ABSTRACT **Data management has emerged as one of the central issues in the high-throughput processes of taking a protein target sequence through to a protein sample. To simplify this task, and following extensive consultation with the international structural genomics community, we describe here a model of the data related to protein production. The model is suitable for both large and small facilities for use in tracking samples, experiments, and results through the many procedures involved. The model is described in Unified Modeling Language (UML). In addition, we present relational database schemas derived from the UML. These relational schemas are already in use in a number of data management projects. Proteins 2005; 58:278–284. © 2004 Wiley-Liss, Inc.**

Key words: UML data model; protein production; database schema; LIMS

INTRODUCTION

As structural biology moves toward developing high-throughput (HTP) pipelines for structure determination,[1–4] data management has become an increasingly central issue. Traditional bookkeeping approaches, such as the laboratory notebook, do not scale to multiuser and multisite environments. It is generally acknowledged that the flow of information within and between different projects is facilitated by storing and managing data electronically. Such considerations have already led the Protein Data Bank[5,6] (PDB) to develop a macromolecular Crystallographic Information File (mmCIF)[7] model that describes the data items involved in the process of solving a protein structure. Software tools, which automatically harvest these data items during the process, greatly simplify the final deposition of the structure with the PDB. For protein structure refinement details, the Crystallography & NMR System (CNS)[8] and Collaborative Computational Project, Number 4 (CCP4)[9,10] software packages are used to generate deposition details in a routine manner.

One way for electronic information to be shared is to define a single data format that can be used to transfer information between software packages.[11,12] An alternative approach is to define the classes of the information and identify relationships between them to produce a structured description of the data of interest—a data model. This approach has been used by the PDB[5,13] and CCPN[14] projects. The data model can then be used to create consistent representations in a number of different formats. It can also be used to automatically generate library code to manipulate the data, simplifying the development and maintenance of software applications that

Grant sponsor: European Union (EU) for the TEMBLOR; Grant number: QLRI-CT-2001-00015. Grant sponsor: SPINE; Grant number: QLG2-CT-2002-00988. Grant sponsor: Biotechnology and Biological Sciences Research Council (BBSRC) UK for the CCPN and eHTPX projects. Grant sponsor: BBSRC through the CCP4 grant (to Alun W. Ashton). Grant sponsor: CCLRC Industrial Placement Studentship (to Peter Wood). Grant sponsor: Medical Research Council, UK (funding for the Oxford Protein Production Facility). Grant sponsor: EU MAXINF; Grant number EU-HPRI-CT-2000-40021 (travel funds for coordinating activities). Grant sponsor: Autostruct projects; Grant number: QLRI-CT-2000-30398 (travel funds for coordinating activities). Grant sponsor: The RCSB (HB & JW) is supported by grants from the USA agencies, the National Science Foundation, the Department of Energy, and the National Institutes of Health.

*Correspondence to: Kim Henrick, Macromolecular Structures Database, EMBL-EBI, The Wellcome Trust Genome Campus, Hinxton, Cambridge CB10 1SD, UK. E-mail: henrick@ebi.ac.uk

Received 22 March 2004; Accepted 13 July 2004

Published online 23 November 2004 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/prot.20303

interact with the data, such as a Laboratory Information Management System (LIMS).[15]

The need for a minimum set of information to describe protein production and to facilitate data exchange was recognized by the International Task Force on Deposition, Archiving, and Curation of Primary Information for Structural Genomics[16] formed in 2001. The task force had representation from the major structural genomics projects and formed some smaller working groups with expertise in the various aspects of the experimental procedures. The task force defined a set of data items that describe protein production and recommended that the structural genomics initiatives collect this information for both successful and unsuccessful experiments. The detailed definitions were further refined at a series of international workshops, including a Structural Genomics Informatics and Software Integration Workshop[17] in 2002 and the Protein Structure Initiative (PSI) Data Management Workshop[18] held in July 2003. This work resulted in an mmCIF data dictionary that allows a description of the protein production process to be included in the PDB deposition of a protein structure. We have extended these definitions to cope with the processing steps involved in producing a protein sample. This was carried out by an iterative interaction with the structural genomics partners of the Structural Proteomics in Europe (SPINE) project[19] and learning from the experience of the Oxford Protein Production Facility.[20]

Although this work is aimed primarily at laboratories developing HTP processes, it should also benefit laboratories carrying out traditional hypothesis-driven research. Anecdotal evidence suggests that large amounts of data are either lost or rendered useless by poor bookkeeping. The adoption of precise electronic standards should lead to better bookkeeping which, as well as being useful in its own right, will also help small laboratories interact efficiently with large-scale facilities.

DESIGN OF THE DATA MODEL

The central idea of the protein production data model (PPDM) is to describe the data required to both reproduce the *samples* and *experiments* involved in protein production and to inform subsequent work. In addition, protein production work is generally directed toward investigating a particular protein, commonly called the *target*. The work often aims to produce a derivative of the target, such as a single domain, to simplify the task or focus on a specific property of the target. We refer to this derivative as the *experimental objective* (an ExpBlueprint in the PPDM). The PPDM is designed around the themes of sample, experiment, target, and experimental objective.

The mmCIF protein production dictionary developed for the PDB was taken as the starting point. Ideas from a number of projects were considered. These include the Sesame Project Management System for Structural Genomics, a web-based software package[21] designed to organize and record data relevant to complex scientific projects, launch computer-controlled processes, and help users make decisions about subsequent steps on the basis of the available information; the Northeast Structural Genomics Consortium hub LIMS system[22] that handles information related to the progress of the consortium when cloning, expressing and purifying proteins; The BASE LIMS[23]; LabRat,[24] BioJava-LIMS,[25] and LabBase[26] systems for sequence and array data; HalX (see below); and the web-based protein crystallography notebook systems Xtrack[27] and LISA.[28]

We identified several key capabilities required of the data model.

- The model must enable the description of the following:

(1) the chemical composition of a sample
(2) the physical location of a sample, including the history of its locations
(3) the involvement of a sample in an experiment
(4) experiment protocols, including the use of robotic systems
(5) experiment results
(6) the sequence of work performed to produce a sample
(7) the relationship between sample, target, and experimental objective
(8) the ownership of samples and experiments

- The model must be sufficiently flexible to cope with unexpected products from experiments.
- The model must be extensible and maintainable.

The PPDM is a formal representation of the protein scientists' understanding of the physical world.[29] It is an abstract description of the relevant data and their relationships implemented in the Unified Modeling Language (UML).[30] UML is an industry-standard, object-oriented modeling language that provides a robust framework for the specification, visualization, and documentation of models. In this work, we made particular use of UML's class diagrams to allow us to describe the samples, experiments, targets, and experimental objectives.

We have developed the UML model within a framework provided by the CCPN project. In that project, information is stored in a UML model from which extensible markup language (XML)[31] schemas, structured query language (SQL) representations, Python and Java application program interfaces (APIs), and documentation can be generated automatically (Fogh et al., manuscript in preparation). Various different tools, such as graphical user interfaces or stand-alone applications, make use of the APIs to manipulate data represented in both XML and SQL. In this way, the data model facilitates flexible software implementation, while ensuring efficient and effective communication between different implementations.

DETAILED DESCRIPTION OF THE PROTEIN PRODUCTION DATA MODEL

The CCPN data model for NMR spectroscopy (Vranken et al., manuscript in preparation) provides a consistent framework for representing and handling information

TABLE I. Packages Involved in the Protein Production Data Model

Protein Production Packages (PPDM)		Shared Packages	
Experiment	Sample	Annotation	Instrument
ExpBlueprint	SampleComponent	ChemComp	Method
Location	Target	Citation	Molecule
RefContainer	Taxonomy	DbRef	People

The PPDM shares the indicated packages with the CCPN data model.

that is useful beyond NMR spectroscopy. To develop this framework in a maintainable way, classes that the data model showed to be closely related were grouped into packages. Each package describes a set of information that can be shared with other packages. We built the PPDM in similar fashion, allowing it to make use of key packages from the CCPN data model. Table I shows the packages involved in the PPDM.

Detailed UML schemas are available at http://www.ebi.ac.uk/msd-srv/docs/ehtpx/lims/. A simplified form of the UML schema is shown in Figure 1 (the Location, RefContainer, and Taxonomy packages have been omitted for clarity). Below, we discuss each of the PPDM packages shown in Figure 1.

ExpBlueprint (Medium Blue Package in Fig. 1)

The experiment blueprint (ExpBlueprint) is used to define an experimental objective; it is the bioinformatical description of a molecular complex that is a target of investigation. This is a more general approach than identifying an objective using a single gene [or open reading frame (ORF)] identifier, as is currently common practice.

An ExpBlueprint can be thought of as a particular combination of polymer and nonpolymer molecules that you wish to study. In terms of the PPDM, this means that each ExpBlueprint is composed of one or more components (BlueprintComponent) that can be either polymers (PolyBlueprintComp), such as DNA or polypeptide, or nonpolymers (NonPolyBlueprintComp). Many polymer BlueprintComponents may be linked back to a single target, representing the fact that different fragments of the same Target may be studied in various contexts.

In order to study an objective (ExpBlueprint), a number of experiments will be performed. ExpBlueprint can therefore also be seen as way of logically grouping experiments. This is represented in the model as an association that links each ExpBlueprint with an arbitrary number of Experiments.

Target (Dark Blue Package in Fig. 1)

The term *target* is widely used within the structural genomics field and has come to represent a number of closely related concepts. Within the PPDM, a Target is defined as the product of a single gene. This description includes the ability to link Targets to information stored in external databases such as SWISS-PROT.[32] In general, it is envisaged that a Target should represent the longest possible product of a particular gene; particular lines of investigation on subdomains should be described at the level of PolyBlueprintComp, as described above.

Sample (Green Package in Fig. 1)

The Sample package contains the description of individual samples, individual holders, and information for tracking them. *Sample* is another term that is commonly overloaded. Within the context of the PPDM, the term *sample* refers to a single piece of labware containing any reagents and to one of two related concepts, distinguished by the value of the isTemplate attribute. Where isTemplate=true (template samples): a generic description of a type of sample (e.g., 1*M* NaCl). Such a sample would be expected to have a list of components but maybe no Holder or Location. Where isTemplate=false (physical samples): a specific description of a particular sample. In addition to a list of components, such a sample would also be expected to have a Holder and a history of Locations. The use of the isTemplate flag is a pragmatic approach to circumvent the extremely complex inheritance hierarchies that would be required to model this area explicitly.

Data tracking of physical samples is critical to laboratory management. This is represented in the PPDM by Holders and Locations (not shown in Fig. 1.). A physical sample is usually held within a container of some form, which in turn may be held within another container, and so on. These relationships are represented in the PPDM by a link between Sample and Holder, and a recursive link between Holder and itself, respectively. The term *holder* refers to any piece of labware that can contain many samples. Each Holder may also have a history of locations associated with it.

SampleComponent (Light Green Package in Fig. 1)

The sample component package contains the description of the sample components that are referred to from the sample package. One of the difficulties when modeling this area is that, to be useful in practice, it must be possible to describe sample components at any one of a number of different levels of detail. For instance, while it is essential to be able to describe certain components in enormous

Fig. 1. A simplified form of the UML protein production data model. Only the major features are represented. Colors denote the packages which are the target gene product (Target, dark blue), the experimental objective (ExpBlueprint, medium blue), samples (Sample, green), sample components (SampleComponent, light green) and experiments performed (Experiment, orange). Samples are tracked by the classes Holder (green) and Location (not shown). Black lines represent relationships between two classes that can indicate one- or two-way navigation, depending on the presence of an arrow showing the direction of the navigability. Multiplicity is specified at the respective end of the association, where an asterisk (*) is used to represent any number. The role of the association is represented by a name at either or both ends (e.g., "+component"). The instance of SampleComponent.AbstractComponent is identified as "component" within one Sample.SampleComponent. Diamonds represent a composition association where the containing class is on the diamond side of the line (e.g., one instance of Sample.Sample can contain many Sample.SampleComponent entities). Orange lines show inheritance between classes (e.g., SampleComponent.AbstractComponent is the super class of all subtyped components). Complete UML class diagrams are available from /www.ebi.ac.uk/msd-srv/docs/ehtpx/lims/.

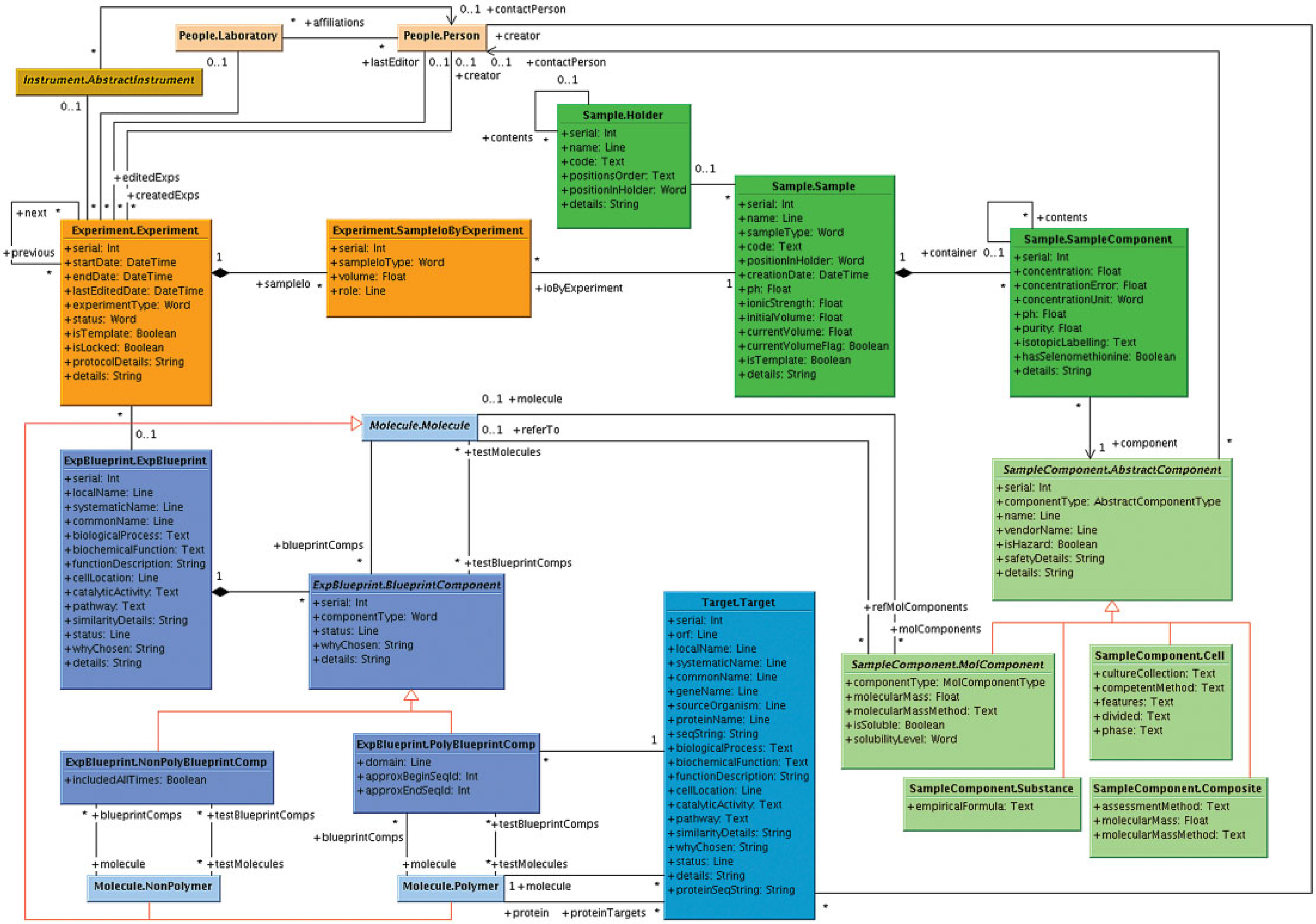
0..1 +contactPerson
People.Laboratory
* +affiliations
People.Person
+creator
Instrument.AbstractInstrument
0..1
* +lastEditor
0..1
0..1 +creator
0..1 +contactPerson
0..1
+contents *
Sample.Holder
+serial: Int
+name: Line
+code: Text
+positionsOrder: Text
+positionInHolder: Word
+details: String
+editedExps
+createdExps
+next *
+previous *
Experiment.Experiment
+serial: Int
+startDate: DateTime
+endDate: DateTime
+lastEditedDate: DateTime
+experimentType: Word
+status: Word
+isTemplate: Boolean
+isLocked: Boolean
+protocolDetails: String
+details: String
1
+sampleIo *
Experiment.SampleIoByExperiment
+serial: Int
+sampleIoType: Word
+volume: Float
+role: Line
+ioByExperiment
Sample.Sample
+serial: Int
+name: Line
+sampleType: Word
+code: Text
+positionInHolder: Word
+creationDate: DateTime
+ph: Float
+ionicStrength: Float
+initialVolume: Float
+currentVolume: Float
+currentVolumeFlag: Boolean
+isTemplate: Boolean
+details: String
* +contents
+container 0..1
Sample.SampleComponent
+serial: Int
+concentration: Float
+concentrationError: Float
+concentrationUnit: Word
+ph: Float
+purity: Float
+isotopicLabelling: Text
+hasSelenomethionine: Boolean
+details: String
1 +component
0..1 +molecule
Molecule.Molecule
0..1 +referTo
* +testMolecules
ExpBlueprint.ExpBlueprint
+serial: Int
+localName: Line
+systematicName: Line
+commonName: Line
+biologicalProcess: Text
+biochemicalFunction: Text
+functionDescription: String
+cellLocation: Line
+catalyticActivity: Text
+pathway: Text
+similarityDetails: String
+status: Line
+whyChosen: String
+details: String
+blueprintComps *
* +testBlueprintComps
ExpBlueprint.BlueprintComponent
+serial: Int
+componentType: Word
+status: Line
+whyChosen: String
+details: String
SampleComponent.AbstractComponent
+serial: Int
+componentType: AbstractComponentType
+name: Line
+vendorName: Line
+isHazard: Boolean
+safetyDetails: String
+details: String
Target.Target
+serial: Int
+orf: Line
+localName: Line
+systematicName: Line
+commonName: Line
+geneName: Line
+sourceOrganism: Line
+proteinName: Line
+seqString: String
+biologicalProcess: Text
+biochemicalFunction: Text
+functionDescription: String
+cellLocation: Line
+catalyticActivity: Text
+pathway: Text
+similarityDetails: String
+whyChosen: String
+status: Line
+details: String
+proteinSeqString: String
+refMolComponents
+molComponents
SampleComponent.MolComponent
+componentType: MolComponentType
+molecularMass: Float
+molecularMassMethod: Text
+isSoluble: Boolean
+solubilityLevel: Word
SampleComponent.Cell
+cultureCollection: Text
+competentMethod: Text
+features: Text
+divided: Text
+phase: Text
ExpBlueprint.NonPolyBlueprintComp
+includedAllTimes: Boolean
ExpBlueprint.PolyBlueprintComp
+domain: Line
+approxBeginSeqId: Int
+approxEndSeqId: Int
SampleComponent.Substance
+empiricalFormula: Text
SampleComponent.Composite
+assessmentMethod: Text
+molecularMass: Float
+molecularMassMethod: Text
* +blueprintComps
* +testBlueprintComps
+blueprintComps
*+testBlueprintComps
+molecule
* +testMolecules
+molecule
*+testMolecules
Molecule.NonPolymer
Molecule.Polymer
1 +molecule
+protein
+proteinTargets *

Figure 1.

detail, it is not helpful to require a detailed chemical description of every cell lysate. This is achieved by careful subtyping of the AbstractComponent class that allows a component to be defined as follows: a molecular component with a well-defined topology (MolComponent); a Cell; a component whose precise topology is not defined or not important (Substance); or some combination of components of these types (Composite). MolComponent is itself subtyped to allow the description of polymer components, such as DNA and polypeptides, and nonpolymer components. Polymers are further subtyped to distinguish restriction enzymes, polymerase chain reaction (PCR) primers, and plasmids from other components (classes not shown in Fig. 1). The SampleComponent package uses the CCPN Molecule package to enable the description of the precise chemistry and stereochemistry of a MolComponent.

A molecule component (MolComponent) is an entity with a well-defined topology; its sequence is stored in Molecule ("+molecule" link). A MolComponent may also have a "referTo" link to a Molecule; this is used to indicate that MolComponents were made in the context of producing a molecule with a given sequence. For instance, DNA or RNA molecules or tagged proteins would "referTo" the protein they were used to produce. A Molecule (be it a Polymer or a Nonpolymer) may be linked to an appropriate BlueprintComponent through the "testMolecules" link; this indicates that the molecule is produced as an attempt to represent the BlueprintComponent itself. This would be the case, for example, for truncation mutants, selenomethionine labeled proteins, and other variants of a target sequence.

Examples of components that would fall under the Substance type are NaCl, characterized by its empirical formula, or fetal calf serum, characterized in practice by its provenance.

Experiment (Orange Package in Fig. 1)

By the strictest definition within the PPDM, an Experiment is defined as any process that creates and/or uses one or more Sample entities. As for Sample, experiment can describe one of two slightly different concepts distinguished by the isTemplate attribute. Where isTemplate=true (template experiment): a generic description of a type of an experiment. This is very close to what is commonly referred to as a protocol and is used to store general methods for reuse. It would not be expected to have values for startDate or endDate, and would be unlikely to refer to large numbers of physical samples. Where isTemplate=false (experiment): a specific description of a particular experiment. Such experiments would be expected to have values for startDate and/or endDate, and would be likely to refer to at least one physical sample.

The precise relationship between template and physical samples, and template and nontemplate experiments is very complex, and no attempt has been made to model this area explicitly for now.

The Experiment class has many subtypes that are used to describe specific Experiment types such as PCR, ligation, expression, and chromatography (classes not shown on Fig. 1). These types match the different *experiments* that can be described using the mmCIF protein production data model. The model can be extended to cover other types of *experiment* by deriving additional Experiment subclasses. In addition, the types of components that are expected in the input and output Samples for each type of Experiment have been analyzed in detail (not shown here but available in the online documentation at http://www.ebi.ac.uk/msd-srv/docs/ehtpx/lims/documentations/html).

Experiments create and use Samples, and details of this relationship are held in SampleIoByExperiment. As well as annotating the experiment, it provides a mechanism for auditing sample volumes. While the typical Experiment will take one or more Samples and produce a new one, there are several other cases worth noting. Sample reception can be described as an Experiment with output but no input Samples. Nondestructive Experiments, such as optical density measurement, can be represented by assigning the same Sample to be both input and output. Destructive Experiments, such as mass spectrometry, can be described as Experiments with input but no output Samples. A Sample output from one Experiment may be an input to another. This directional relationship will intrinsically describe the workflow that produced the protein.

AN SQL IMPLEMENTATION

The UML data model is an abstract representation of the structure of the data. It neither stores nor provides access to the data. The storage function is best provided by relational database management systems (RDBMS) such as Oracle,[33] PostgreSQL,[34] and mySQL.[35] The common language of RDBMS is SQL. Several rules are used to transform the UML model into an SQL schema. First, each class in the UML model is represented as a table with a column list matching the attributes of the class. Second, for each many-to-one or one-to-one association of that class, a foreign key is created. Further, if the association is many-to-many, an additional table is created containing two foreign keys that point to the main tables. Third, for inheritance association, two of the most common strategies are used here: Either all classes of the inheritance tree are mapped onto one table, or each class of the inheritance tree is mapped to a distinct table without mapping base class fields to derived classes. In the latter case, a foreign key is created for the inheritance association. The CCPN framework provides additional information to direct the transformation.

Unfortunately, each RDBMS has its own particular dialect of SQL. To allow for the widest choice of RDBMS, the UML is first transformed into XML that describes the structure of the database. The XML is processed by Torque[36] to generate SQL specific to different RDBMS. We support the SQL schemas specific to Oracle, PostgreSQL, and mySQL. Detailed documentation for each table and each field is automatically generated by Torque from documentation located in the UML data model.

APPLICATIONS

The CCPN framework also provides the facility to generate Python and Java APIs (mechanism to be described separately; Fogh et al., manuscript in preparation). The API provides methods to access the underlying RDBMS to store and retrieve data. This allows an application to manipulate data without a detailed knowledge of the way in which the data is stored. This speeds up application development and simplifies maintenance.

The main applications built on the PPDM model are likely to be LIMS. Two applications closely associated with the work described here are HalX[37] and MOLE.[38] MOLE uses a mySQL implementation of the PPDM. HalX (and the BRI-NRC LIMS in Montreal) is currently implementing both the Java API and PostgreSQL schema. Each represents a different implementation: Each system handles data capture in a different manner, using different software technologies, but the intention is that they should be able seamlessly to exchange data.

XML SCHEMA FOR DATA EXCHANGE

XML is a standardized application-independent way of representing structured data and, as such, is rapidly becoming the standard mechanism for data exchange. XML has been used to define data exchange schemas in related areas of research. Target and status information for structural genomics centers worldwide is available as XML from TargetDB (http://targetdb.pdb.org/). An extended version is used by the SPINE initiative to share structural genomics data within Europe. Various other XML exchange standards have been established for large-scale biological data, including for microarray and proteomics data.[39–43] XML is also ideal for the long-term storage of structured data, such as the archiving of analytical instrument data.[44] The PPDM is also transformed into an XML representation using the same technology as described for SQL above.

The eHTPX data pipelining project[45] is closely associated with this work. It allows the user to manage the flow of data from the initial stages of target selection to the automated deposition of the final refined model in the public databases. XML schemas have been developed for the exchange of information between the different stages, including the exchange of protein production data. Grid technology will be used to link the stages (Fillon et al., manuscript in preparation).

CONCLUSIONS

The PPDM has been developed to help laboratory workers and software applications to collect, store, and exchange information concerning protein production through the provision of a common platform. The SPINE initiative and the HalX and MOLE LIMS have already adopted it as a standard. It is hoped that take-up by industry will allow tighter integration with facilities offering commercial services and simplify the task of integrating robotic systems into labs.

AVAILABILITY AND DOWNLOADS

The PPDM is freely available under the LGPL license. The UML, SQL, and documentation are available online (http://www.ebi.ac.uk/msd-srv/docs/ehtpx/lims/downloads.html).

The CCPN data model (incorporating the PPDM) and the Python and Java API's are available online (http://www.ccpn.ac.uk/).

The mmCIF dictionary for protein production is available online (http://deposit.pdb.org/mmcif/sg-data/protprod.html).

REFERENCES

1. Thornton J. Structural genomics takes off. Trends Biochem Sci 2001;26:88–89.
2. Jhoti H. High throughput structural proteomics using X-rays. Trends Biotechnol 2001;19(Suppl):S67–S71.
3. Edwards AM, Arrowsmith CH, Christendat D, Dharamsi A, Friesen JD, Greenblatt JF, Vedadi M. Protein production: feeding the crystallographers and NMR spectroscopists. Nat Struct Biol 2000;7(Suppl):970–962.
4. Sali A, Glaeser R, Earnest T, Baumeister W. From words to literature in structural proteomics. Nature 2003;422:217–225.
5. Berman HM, Westbrook JD, Feng Z, Gilliland G, Bhat TN, Weissig H, Shindyalov IN, Bourne PE. The Protein Data Bank. Nucleic Acids Res 2000;28:235–242.
6. Bernstein FC, Koetzle TF, Williams GJ, Meyer EE, Brice MD, Rodgers JR, Kennard O, Shimanouchi T, Tasumi M. Protein Data Bank: a computer-based archival file for macromolecular structures. J Mol Biol 1977;112:535–542.
7. Bourne P, Berman HM, Watenpaugh K, Westbrook JD, Fitzgerald PMD. The macromolecular Crystallographic Information File (mmCIF). Methods Enzymol 1997;277:571–590.
8. Brunger AT, Adams PD, Clore GM, DeLano WL, Gros P, Grosse-Kunstleve RW, Jiang J-S, Kuszewski J, Nilges M, Pannu NS, Read RJ, Rice LM, Simonson T, Warren GL. Crystallograpy and NMR system (CNS): a new software system for macromolecular structure determination. Acta Crystallogr D Biol Crystallogr 1998;54: 905–921.
9. Collaborative Computational Project, Number 4. The CCP4 Suite: programs for protein crystallography. Acta Crystallogr D Biol Crystallogr 1994;50:760–763.
10. Potterton E, Briggs P, Turkenburg M, Dodson E. A graphical user interface to the CCP4 program suite. Acta Crystallogr D Biol Crystallogr 2003;59:1131–1137.
11. Achard F, Vayssiex G, Barillot E. XML, bioinformatics and data integration. Bioinformatics 2001;17:115–125.
12. Brazma A. On the importance of standardisation in life sciences. Bioinformatics 2001;17:113–114.
13. Westbrook J, Bourne PE. STAR/mmCIF: an extensive ontology for macromolecular structure and beyond. Bioinformatics 2000;16: 159–168.
14. Fogh RH, Ionides JMC, Ulrich E, Boucher W, Vranken W, Linge JP, Habeck M, Rieping W, Bhat TN, Westbrook J, Henrick K, Gililand G, Berman H, Thornton J, Nilges M, Markley J, Laue ED. The CCPN project: an interim report on a data model for the NMR community. Nat Struct Biol 2002;9:416–418.
15. Avery G, McGee C, Falk S. Implementing LIMS: a "how-to" guide. Anal Chem 2000;72:57A–62A.
16. Second International Structural Genomics Meeting, April 4–6, 2001, Airlie, VA. Available online at http://www.nigms.nih.gov/news/meetings/airlie.html
17. Structural Genomics Informatics and Software Integration Workshop (SG ISI), Hyatt Regency, San Antonio, TX, May 24–25, 2002. Available online at http://deposit.pdb.org/sgisi02/
18. Protein Structure Initiative Data Management Workshop, July 10–11, 2003, at the National Institutes of Health, in Bethesda, MD. Available online at http://www.nigms.nih.gov/psi/meetings/data_management.html
19. Structural Proteomics in Europe. Available online at http://www.spineurope.org/
20. Walter TS, Diprose J, Brown J, Pickford M,Owens RJ, Stuart DI, Harlos K. A procedure for setting up high-throughput, nanolitre crystallization experiments: I. Protocol design and validation. J Appl Crystallogr 2003;36:308–314. Available online at http://www.oppf.ox.ac.uk/
21. Zolnai Z, Lee PT, Li J, Chapman MR, Newman CS, Phillips GN Jr,

Rayment I, Ulrich EL, Volkman BF, Markley JL. Project management system for structural and functional proteomics: SESAME. J Struct Funct Genomics 2003;4:11–23. Available online at http://www.sesame.wisc.edu

22. Goh C-S, Lan N, Echols N, Douglas SM, Milburn D, Bertone P, Xiao R, Ma L-C, Zheng D, Wunderlich Z, Acton T, Montelione GT, Gerstein M. SPINE 2: a system for collaborative structural proteomics within a federated database framework. Nucleic Acids Res 2003;31:2833–2838.
23. Saal LH, Troein C, Vallon-Christersson J, et al. BioArray Software Environment (BASE): a platform for comprehensive management and analysis of microarray data. Genome Biol 2002;3(8): Software. Available online at http://base.thep.lu.se/
24. Breese MR, Stephens MJ, McClintick JN, Grow MW, Edenberg HJ, Labrat HJ. LIMS: an extensible framework for developing laboratory information management, analysis, and bioinformatics solutions for microarrays. Available online at http://d1.cmg.iupui.edu/~marcus/cmg/software.html
25. Rolfe PA. BioJava LIMS. Available online at http://cvs.bioperl.org/cgi-bin/viewcvs/viewcvs.cgi/biojava-lims/?cvsroot=biojava
26. Rozen S, Stein L, Goodman N. (1995) LabBase: managing lab data in a large-scale genome-mapping. IEEE Eng Med Biol Mag 1995;14:702–709. Available online at http://www.broad.mit.edu/ftp/distribution/software/labbase/
27. Harris M, Jones TA. Xtrack—a web-based crystallographic notebook. Acta Crystallogr D Biol Crystallogr 2002;58:1889–1891.
28. Haebel PW, Arcus VL, Baker EN, Metcalf P. LISA: an intranet-based flexible database for protein crystallography project management. Acta Crystallogr D Biol Crystallogr 2001;57:1341–1343.
29. Meyer B. Object oriented software construction. 2nd edition. Prentice-Hall; 1997.
30. Unified Modeling Language. Available online at http://www.uml.org/
31. Extensible Markup Language. Available online at http://www.xml.org/
32. Bairoch A, Apweiler R. The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000. Nucleic Acids Res 2000;28:45–48.
33. Oracle. Oracle Corporation; Available online at http://www.oracle.com
34. The PostgreSQL Global Development Group Available online at http://www.postgresql.org
35. MySQL AB. Uppsala, Sweden. Available online at http://www.mysql.com
36. The Apache DB Project. Available online at http://db.apache.org/torque/
37. Poupon A. HALX. Available online at http://halx.genomics.eu.org/
38. Morris C, Wood P, Griffiths SL, Wilson KS, Ashton AW. MOLE: a data management application based on a protein production data model. Proteins 2004. Forthcoming. See also http://www.mole.ac.uk
39. Brazma A, Hingamp P, Quackenbush J, Sherlock G, Spellman P, Stoeckert C, Aach J, Ansorge W, Ball CA, Causton HC, Gaasterland T, Glenisson P, Holstege FC, Kim IF, Markowitz V, Matese JC, Parkinson H, Robinson A, Sarkans U, Schulze-Kremer S, Stewart J, Taylor R, Vilo J, Vingron M. Minimum information about a microarray experiment (MIAME)—towards standards for microarray data. Nat Genet 2001;29:365–371.
40. Spellman PT, Miller M, Stewart J, Troup C, Sarkans U, Chervitz S, Bernhart D, Sherlock G, Ball C, Lepage M, Swiatek M, Marks WL, Goncalves J, Markel S, Iordan D, Shojatalab M, Pizarro A, White J, Hubley R, Deutsch E, Senger M, Aronow BJ, Robinson A, Bassett D, Stoeckert CJ Jr, Brazma A. Design and implementation of microarray gene expression markup language (MAGE-ML). Genome Biol 2002;3(9):RESEARCH0046. Available online at http://www.mgu.har.mrc.ac.uk/facilities/microarray/, http://www.mged.org/Workgroups/MAGE/, and http://www.ebi.ac.uk/arrayexpress/Standards/MAGE.
41. Taylor CF, Paton NW, Garwood KL, Kirby PD, Stead DA, Yin Z, Deutsch EW, Selway L, Walker J, Riba-Garcia I, Mohammed S, Deery MJ, Howard JA, Dunkley T, Aebersold R, Kell DB, Lilley KS, Roepstorff P, Yates JR 3rd, Brass A, Brown AJ, Cash P, Gaskell SJ, Hubbard SJ, Oliver SG. A systematic approach to modelling, capturing and disseminating proteomics experimental data. Nat Biotechnol 2003;21:247–254. Available online at http://pedro.man.ac.uk.
42. Orchard S, Hermjakob H, Apweiler R. The Proteomics Standards Initiative. Proteomics 2003;3:1374–1376.
43. Hermjakob H, Montecchi-Palazzi L, Bader G, Wojcik J, Salwinski L, Ceol A, Moore S, Orchard S, Sarkans U, von Mering C, Roechert B, Poux S, Jung E, Mersch H, Kersey P, Lappe M, Li Y, Zeng R, Rana D, Nikolski M, Husi H, Brun C, Shanker K, Grant SG, Sander C, Bork P, Zhu W, Pandey A, Brazma A, Jacq B, Vidal M, Sherman D, Legrain P, Cesareni G, Xenarios I, Eisenberg D, Steipe B, Hogue C, Apweiler R. The HUPO PSI's molecular interaction format—a community standard for the representation of protein interaction data. Nat Biotechnol 2004;22:177–183. See http://psidev.sourceforge.net/
44. Generalized Analytical Markup Language for analytical instrumentation. Available online at http://www.gaml.org/
45. Allan R, Diakun G, Guest M, Keegan R, Nave C, Papiz , Winn M, Winter G, Diprose J, Esnouf R, Mayo C, Stuart D, Launer L, Walsh, M, Fillon J, Henrick K, Pajon A, Cowtan K, Young P, Randy , Read R, Rana O. An e-science resource for high throughput protein crystallography. Available online at http://www.e-htpx.ac.uk/