Pacific Symposium on Biocomputing 7:88-99 (2002)

AUTOMATING DATA ACQUISITION INTO ONTOLOGIES FROM PHARMACOGENETICS RELATIONAL DATA SOURCES USING DECLARATIVE OBJECT DEFINITIONS AND XML

DANIEL L. RUBIN, MICHEAL HEWETT, DIANE E. OLIVER, TERI E. KLEIN,
AND RUSS B. ALTMAN

Stanford Medical Informatics, MSOB X-215,
Stanford, CA 94305-5479 USA
E-mail: rubin@smi.stanford.edu, altman@smi.stanford.edu

Ontologies are useful for organizing large numbers of concepts having complex relationships, such as the breadth of genetic and clinical knowledge in pharmacogenomics. But because ontologies change and knowledge evolves, it is time consuming to maintain stable mappings to external data sources that are in relational format. We propose a method for interfacing ontology models with data acquisition from external relational data sources. This method uses a declarative interface between the ontology and the data source, and this interface is modeled in the ontology and implemented using XML schema. Data is imported from the relational source into the ontology using XML, and data integrity is checked by validating the XML submission with an XML schema. We have implemented this approach in PharmGKB (http://www.pharmgkb.org/), a pharmacogenetics knowledge base. Our goals were to (1) import genetic sequence data, collected in relational format, into the pharmacogenetics ontology, and (2) automate the process of updating the links between the ontology and data acquisition when the ontology changes. We tested our approach by linking PharmGKB with data acquisition from a relational model of genetic sequence information. The ontology subsequently evolved, and we were able to rapidly update our interface with the external data and continue acquiring the data. Similar approaches may be helpful for integrating other heterogeneous information sources in order make the diversity of pharmacogenetics data amenable to computational analysis.

1 Introduction

1.1 Pharmacogenetics and the need to connect diverse data

Connecting genotype and phenotype data is the quest of pharmacogenetics[*]—a discipline that seeks to understand how inherited genetic differences among people influence their response to drugs. Discovering important relationships between genes and drugs could lead to personalized medicine, where drug therapy is customized according to the genetic constitution of the patient. Thus, there is great interest in rapidly acquiring genotype and phenotype data in many individuals, and clinical trials in the future will routinely collect genotype as well as phenotype information.[1]

Modern experimental methods such as high-throughput DNA sequencing techniques and gene-expression microarrays are contributing detailed genetic and phenotypic information at a rapid rate.[2,3] These abundant and diverse data are a rich source for developing a comprehensive picture of relationships among genes and

[*] We will consider the term “pharmacogenomics” to be equivalent to “pharmacogenetics.”

drugs, but they also create new and complex problems for data integration and interpretation. The plethora of diverse databases having genomic,[4-7] cellular,[8] and phenotype information[9] exacerbates this complexity. Even within a given class of database, such as those containing genetic sequence data, the organization, terminologies, and data models differ.[6,7,10] It is difficult to integrate heterogeneous databases, and standards are not easily adopted.[3]

In response to the need for an integrated resource for pharmacogenetics research, the National Institutes of Health funded the Pharmacogenetics Research Network and Knowledge Base initiative, including the pharmacogenetics knowledge base (PharmGKB).[11] The goal of the PharmGKB project is to develop a knowledge base that can become a national resource containing high quality publicly-accessible pharmacogenetics data that connects genotype, molecular/cellular phenotype, and clinical phenotype information. The challenge for PharmGKB is to integrate a wide scope of genetic and phenotypic information.

1.2 Integrating data in ontologies

To integrate diverse genetic, cellular phenotypic, and clinical information, it is necessary to develop a data model that specifies the pertinent concepts, the semantics of these concepts, and the relationships among them. Because biological understandings evolve, and new types of information continue to emerge after a database design is established, the data model changes. However, when the data model changes, the links to outside sources of data must be updated, which can be a time-consuming process.

Ontologies are models that describe concepts and the relationships among them, combining an abstraction hierarchy of concepts with a semantic network of relationships. Ontologies are flexible and highly expressive, and have been useful for building knowledge bases in biology,[12-15] as well as in the PharmGKB project.[16]

A disadvantage of ontologies is that network and hierarchical data models are very different from flat tabular relational models, and ontologies are not easily integrated with relational data sources; yet the latter are predominant in most biology databases[4,7,17] and experimental laboratories today. This is not a problem when the ontologies are relatively stable, do not change once data acquisition begins, and are manually curated to ensure integrity of the data.[14,15] But while developing the ontology for PharmGKB, it became clear that it will continue to change as our understanding of the concepts and relationships in pharmacogenetics data evolves. Furthermore, many biomedical scientists think about their data in terms of tables (a relational view), not in terms of ontologies. Our challenge, therefore, is to develop a robust interface between relational data acquisition and the PharmGKB ontology. We also sought a method that would automate updating this interface when the ontology changes.

1.3 XML and data exchange

Extensible Markup Language (XML[18]) is useful as a data representation scheme[19-21] and for exchanging data between resources and databases.[22-24] XML provides a general framework for exchanging data between resources because it is extensible, readable by humans, unambiguously parsed by computers, and can be formally defined using a document type definition (DTD) or XML schema. XML schema[25] is a more powerful language for defining XML formats. XML schema is superior to a DTD for expressing constraints because XML schemas specify not only the structure but also the data type of each element and attribute. XML schemas are written in XML, and thus are self-describing and easier to understand than a DTD. XML schemas are also extensible, permitting authors to develop customized constraints.

Data integration requires access to a variety of data sources through a single mediated schema. A major difficulty with integrating data from outside sources is the laborious manual construction of semantic mappings between the source schemas and the mediated schema. It is also necessary to validate the incoming data against the legal ranges for each field in the importing database. If we were to develop an XML schema to serve as the mediating schema, this would address the problem of validating the structure and content of incoming data. But we would still need to have a way of defining the content in the XML schema. Ideally, the XML schema should be defined from information in the PharmGKB ontology. We have developed a method for using an ontology to define a mediating XML schema.

2 Method

2.1 Overview of our method

Our method consists of several components that are shown schematically in Figure 1. The first component is the PharmGKB ontology, which contains the concepts (classes) that describe the domain of pharmacogenetics, and it also models the relationships among the classes (Figure 2, left side). Data are stored in the ontology by creating instances of these classes and storing the data in the appropriate slots (named attributes that store data) associated with the instances. To specify a relationship between instances, we connect them by assigning one instance to the slot value in the other instance. For example, a PCR assay submission has relationships to two instances: a forward PCR primer and a reverse PCR primer (Figure 2, right side). This relationship allows us to specify the particular primers used in a PCR assay.

The second component of our system is the XML schema (Figure 1), which is derived from the ontology and used as an interface between data acquisition and the ontology. The ontology contains a declarative representation of data constraints that are used to define validation constraints on incoming data, and to create the XML schema. This component includes an XML parser that validates incoming XML documents against the schema, creates new instances in the ontology and assigns their slot values from data in the parsed XML document, and creates the necessary links among the instances.

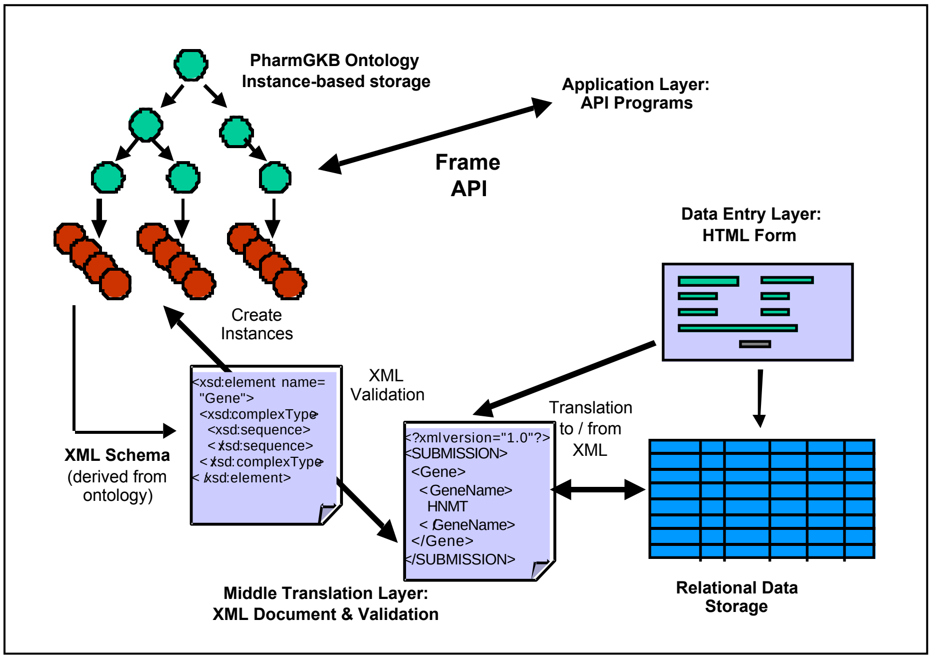
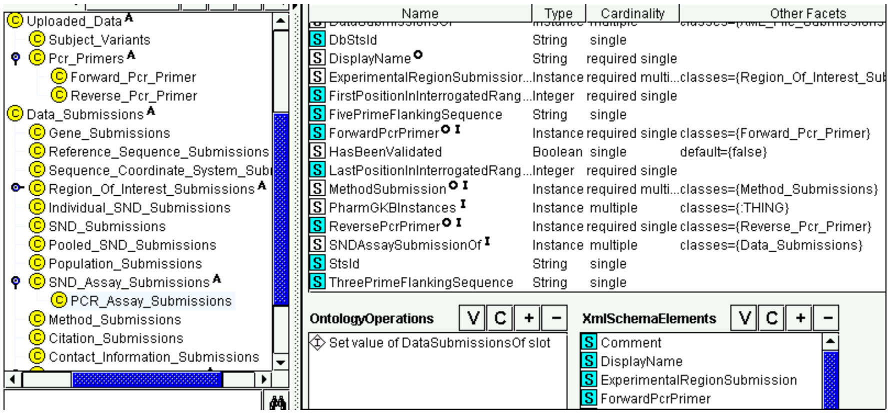


Figure 1. Model for data acquisition in PharmGKB. The PharmGKB ontology (above left) is a network of interrelated classes (upper circles) and instances (lower circles), which store data in slots (not shown). Data to be integrated from an external sources (either web forms or relational schemas) is transmitted in an XML document whose syntax is specified by an XML schema (the latter is derived from the ontology). The data in the XML document is stored in instances in PharmGKB that are created when the document is processed by the XML parsing module.

The third component in our method is an XML translator that converts external incoming relational data from an HTML web form into XML. It is also possible to submit data directly from a relational data source if the data are put into an XML document that is valid against the XML schema.

2.2 Ontology model of genetic information and data validity constraints

We initially developed and refined the PharmGKB ontology of genetic sequence data through a process of iterative refinement, where we evaluated the data currently available in genetic sequence databases as well as sample data from two study centers in the PharmGKB network, built a preliminary model, and subsequently reevaluated and revised the ontology. The ontology was developed using the Protégé suite of tools.[26] Protégé has a graphical user interface for editing ontologies. It is designed for rapidly evolving knowledge bases, which made managing changes to the ontology easier for us. The tool set also made the ontology readily available to application programs that use the ontology.



Uploaded_Data
- Subject_Variants
- Pcr_Primers
  - Forward_Pcr_Primer
  - Reverse_Pcr_Primer

Data_Submissions
- Gene_Submissions
- Reference_Sequence_Submissions
- Sequence_Coordinate_System_Sub
- Region_Of_Interest_Submissions
- Individual_SND_Submissions
- SND_Submissions
- Pooled_SND_Submissions
- Population_Submissions
- SND_Assay_Submissions
  - PCR_Assay_Submissions
- Method_Submissions
- Citation_Submissions
- Contact_Information_Submissions

Name	Type	Cardinality	Other Facets
DbStsId	String	single	
DisplayName	String	required single	
ExperimentalRegionSubmissior...	Instance	required multi...	classes={Region_Of_Interest_Sul
FirstPositionInInterrogatedRang...	Integer	required single	
FivePrimeFlankingSequence	String	single	
ForwardPcrPrimer	Instance	required single	classes={Forward_Pcr_Primer}
HasBeenValidated	Boolean	single	default={false}
LastPositionInInterrogatedRang...	Integer	required single	
MethodSubmission	Instance	required multi...	classes={Method_Submissions}
PharmGKBInstances	Instance	multiple	classes={:THING}
ReversePcrPrimer	Instance	required single	classes={Reverse_Pcr_Primer}
SNDAssaySubmissionOf	Instance	multiple	classes={Data_Submissions}
StsId	String	single	
ThreePrimeFlankingSequence	String	single	

OntologyOperations
- Set value of DataSubmissionsOf slot

XmlSchemaElements
- Comment
- DisplayName
- ExperimentalRegionSubmission
- ForwardPcrPrimer

Figure 2. View of part of the PharmGKB model in the Protégé graphical user interface demonstrating both the ontology and constraints that specify the XML schema. The left panel displays the hierarchy of classes making up the ontology. Each class has slots that store data in the ontology. The slots for the "*PCR_Assay_Submissions*" class are shown (right top panel). Constraints on the values for data submitted are stored with the slots that store that data (*MethodSubmission* is an instance, it is required, and is multiple cardinality; *DbStsId* is a string, single-cardinality, and not required; these constraints are stored with each slot as seen in the right top panel). Some of the slots have values that are instances from other classes; for example, the slot "*ForwardPCRPrimer*" has a range of "*Forward_PCR_Primer*", the latter being another class in the ontology (seen in the top of the left panel). Some of the slots in the ontology are used for administrative purposes; those that are used for data acquisition from outside sources are listed in the lower right panel, "XmlSchemaElements" in the order required in the XML document.

The ontology includes slots that contain data submitted to PharmGKB ("XML schema slots") and slots that are used for internal purposes in the knowledge base ("administrative slots"). For example, in the *PCR_Assay_Submissions* class (Figure 2), the *StsId* slot contains an STS identifier; the *HasBeenValidated* slot is used internally by PharmGKB to ascertain whether existing instances of *PCR_Assay_Submission*s have passed higher-level data validations.

After the ontology was built, we added these declarative constraints to the ontology (they define the XML schema used to validate data submitted to PharmGKB):

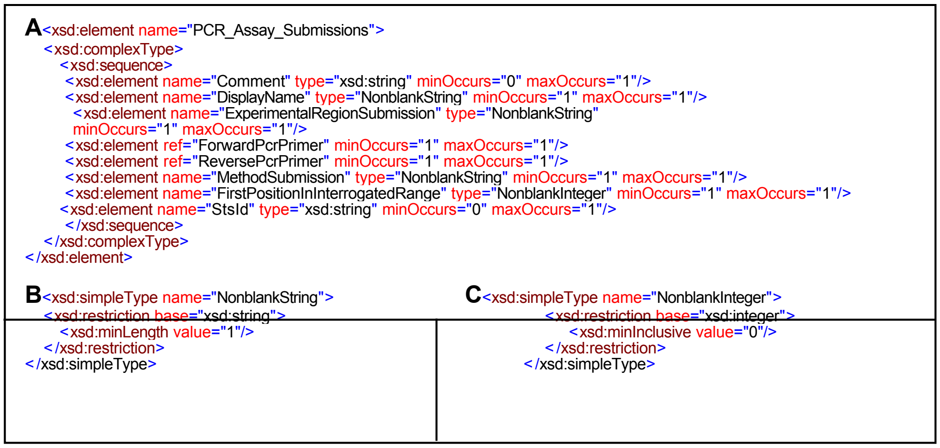
- A list of XML schema slots and the order they are to appear in XML documents
- The required data type for each XML schema slot (integer, string, instance, etc.)
- The cardinality (single or multiple) of each XML schema slot
- A flag indicating if a value is required or optional for each XML schema slot.

Figure 2 (right panel) shows how these constraints are represented in the ontology. Constraints such as data type, cardinality, and whether the data are optional or required are stored with the slot that will contain the corresponding data. Our method uses the following convention for naming XML elements: class and slot names are the same in the XML schema. The names of slots and classes are globally unique in PharmGKB, which prevents naming conflicts. Thus, the ontology in Figure 2 can be interpreted as a declarative representation of an XML schema.

2.3 Creating the XML schema

In order to generate an XML schema from the ontology, there must be a convention for naming and organizing the XML elements and attributes. To preserve the desired close connection between the ontology class/slot structure and the XML schema, we organized the XML schema into a set of nested elements having no attributes. The name of the outermost XML element is always the name of a class in the ontology, and each of a series of sub-elements is given the same name as the corresponding slot in that class. The data being submitted is contained within these sub-elements (Figure 3A).

Once the ontology is built and the constraints on data values are declared, the XML schema is sufficiently determined, and it can be compiled directly from the ontology (Figure 3). There is actually more than one way to write equivalent XML schemas, so we cannot say the XML schema is completely determined. In Figure 3B, for example, specification of the required value constraints could have been placed within the XML schema elements that use them, without needing a separate declaration. The alternative ways of writing the XML schema convey the same constraints, so we assert that to the extent that it encodes constraints on content and data validations, the XML schema is sufficiently determined.



```
A<xsd:element name="PCR_Assay_Submissions">
   <xsd:complexType>
      <xsd:sequence>
       <xsd:element name="Comment" type="xsd:string" minOccurs="0" maxOccurs="1"/>
       <xsd:element name="DisplayName" type="NonblankString" minOccurs="1" maxOccurs="1"/>
        <xsd:element name="ExperimentalRegionSubmission" type="NonblankString"
        minOccurs="1" maxOccurs="1"/>
       <xsd:element ref="ForwardPcrPrimer" minOccurs="1" maxOccurs="1"/>
       <xsd:element ref="ReversePcrPrimer" minOccurs="1" maxOccurs="1"/>
       <xsd:element name="MethodSubmission" type="NonblankString" minOccurs="1" maxOccurs="1"/>
       <xsd:element name="FirstPositionInInterrogatedRange" type="NonblankInteger" minOccurs="1" maxOccurs="1"/>
      <xsd:element name="StsId" type="xsd:string" minOccurs="0" maxOccurs="1"/>
       </xsd:sequence>
   </xsd:complexType>
</xsd:element>
```

```
B<xsd:simpleType name="NonblankString">
   <xsd:restriction base="xsd:string">
      <xsd:minLength value="1"/>
   </xsd:restriction>
</xsd:simpleType>
```

```
C<xsd:simpleType name="NonblankInteger">
         <xsd:restriction base="xsd:integer">
            <xsd:minInclusive value="0"/>
         </xsd:restriction>
      </xsd:simpleType>
```

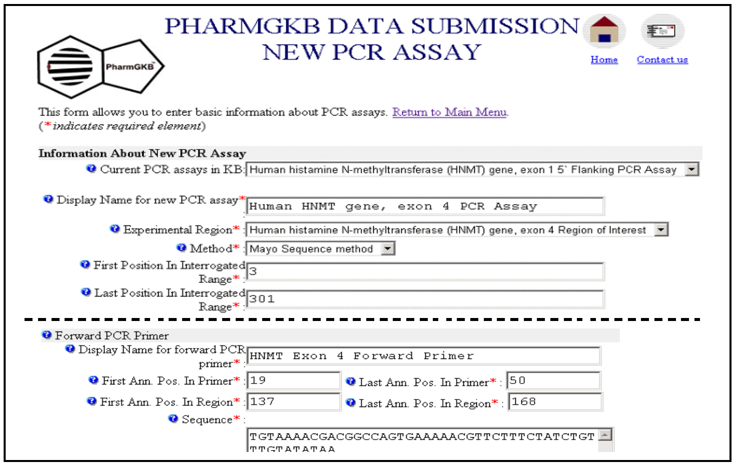
Figure 3. **A**: An excerpt of the XML schema defining the format and constraints for submitting PCR assay data (not all the element definitions are shown). Note that the name of the outermost element matches the name of a class in the ontology (Figure 1, left panel), while the names of the sub-elements match the names of the XML schema slots in the ontology (Figure 1, right panel). For each of these sub-elements, the data type, cardinality, and required/optional status matches that specified in the ontology (Figure 1, right panel). **B, C**: XML schema defining custom data types: a string that must not be blank (B) and an integer value that is required (C).

For this study, we generated the XML schema by copying the content and data constraints from the ontology directly into the XML schema; we are developing a program to generate the XML schema automatically from the ontology. The current

XML schema for PharmGKB is available at http://www.pharmgkb.org/xml-schemas.html.

2.4 Data acquisition

Data acquired from external relational data sources must be put into an XML document that uses the syntax specified by XML schema. Generally, this is a direct mapping from columns in a relational table to the appropriate elements in the XML schema. Because the organization of the XML schema parallels the structure of the ontology, creating an XML document involves collecting the data pertaining to each class in the ontology for which data is to be submitted. For example, to submit data for a PCR assay, a single *PCR_Assay_Submissions* element and its sub-elements are created (Figure 3A), and all the necessary data can be provided in a flat list that is similar to relational structures. Note that some submissions refer to preexisting instances in PharmGKB (e.g., a *PCR_Assay_Submissions* instance refers to forward and reverse primers). All instances have a slot named "*DisplayName*" which is used as the handle to the instance. If the value of an XML element is an instance in PharmGKB, then the *DisplayName* of the instance is provided. If that instance needs to be created at the time of the submission, the data for that instance is provided either by nesting additional elements in the XML file (as is the case for forward and reverse primers in Figure 3A), or as an additional XML element preceding the one that refers to it. In general, relational data to be input into PharmGKB can be directly mapped to a set of XML elements.



PHARMGKB DATA SUBMISSION
NEW PCR ASSAY

This form allows you to enter basic information about PCR assays. Return to Main Menu.
(* *indicates required element*)

Information About New PCR Assay

Current PCR assays in KB:	Human histamine N-methyltransferase (HNMT) gene, exon 1 5` Flanking PCR Assay
Display Name for new PCR assay*:	Human HNMT gene, exon 4 PCR Assay
Experimental Region* :	Human histamine N-methyltransferase (HNMT) gene, exon 4 Region of Interest
Method* :	Mayo Sequence method
First Position In Interrogated Range* :	3
Last Position In Interrogated Range* :	301

Forward PCR Primer

Display Name for forward PCR primer* :	HNMT Exon 4 Forward Primer		
First Ann. Pos. In Primer* :	19	Last Ann. Pos. In Primer* :	50
First Ann. Pos. In Region* :	137	Last Ann. Pos. In Region* :	168
Sequence* :	TGTAAAACGACGGCCAGTGAAAAACGTTCTTTCTATCTGTTTGTATATAA		

Figure 4. Portions of the HTML web form used for submitting PCR assay data to PharmGKB. Data in the form of strings and numbers are directly entered on the form. Values representing instances in PharmGKB are selected from pull-down menus that list all the relevant objects currently in PharmGKB. If a new object needs to be created, there is either a separate web form, or there are additional fields on the same web form for that purpose. Top half of figure shows the top of the form. Lower half of figure shows fields for entering information for new forward and reverse (not shown) PCR primers.

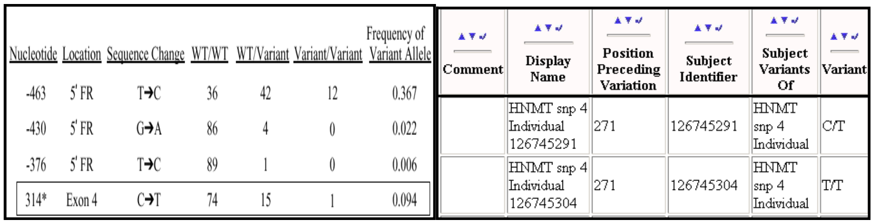
We created a set of HTML data entry forms to simplify the task of entering data into PharmGKB. The types of forms follow the types of classes in the ontology (Figure 1): There are separate forms for submitting genes, sequences, PCR assays, etc. In cases where a submission will create more than one instance in the ontology, all the required fields are supplied on the form. For example, for PCR assay submissions, there are fields for the specifics of the assay (Figure 4, top of figure) as well as for the forward and reverse primers (Figure 4, bottom of figure). The parsing module creates the necessary instances and links them (the instance for the assay is linked to instances for the forward and reverse primers).

2.5 Ontology evolution and propagating changes

The challenge of using an interface is updating it when the ontology changes. Our method automates the process of updating the XML schema interface. In our ontology design, there are two kinds of slots (see section 2.2): XML schema slots and administrative slots. If the change in ontology structure affects only administrative slots, then there will be no change in the XML schema or data acquisition. If the change affects XML schema slots, then a new XML schema must be created. Because the XML schema is directly determined from the ontology, changes to XML schema slots in the ontology can be directly transferred to the XML schema. At the time a new version of the ontology is created, a new version of the XML schema can be immediately produced, so new data can be submitted to PharmGKB using the new version of the XML schema. All XML schemas have a required element that stores a version number so that all incoming XML documents can be identified with respect to version of the schema.

3 Evaluation

We have tested our approach by implementing it in a production system. PharmGKB accepts data from multiple study centers. They submit data either through web forms (Figure 4) or by direct submission of XML files. The study centers provided copies of their raw data; this confirmed that they organize and store their data in a tabular format (Figure 5). We tested the ability of our system to acquire their data by requesting one of the study centers to submit the same data in an XML file. Because the data model in the XML schema is similar to a flat file structure and the XML element names describe the data they contain, tabular relational data was directly translated into XML. In an initial draft of their XML submission, some of the required data values were missing—this was discovered when the XML document was validated against the XML schema.



Nucleotide	Location	Sequence Change	WT/WT	WT/Variant	Variant/Variant	Frequency of Variant Allele
-463	5' FR	T→C	36	42	12	0.367
-430	5' FR	G→A	86	4	0	0.022
-376	5' FR	T→C	89	1	0	0.006
314*	Exon 4	C→T	74	15	1	0.094

▲▼✓ Comment	▲▼✓ Display Name	▲▼✓ Position Preceding Variation	▲▼✓ Subject Identifier	▲▼✓ Subject Variants Of	▲▼✓ Variant
	HNMT snp 4 Individual 126745291	271	126745291	HNMT snp 4 Individual	C/T
	HNMT snp 4 Individual 126745304	271	126745304	HNMT snp 4 Individual	T/T

Figure 5. Display of a summary of the polymorphism data in PharmGKB (right) after importing the data (left). Although this display appears similar to the format of the raw data, the data is actually stored as a set of linked instances in the PharmGKB ontology. This is a partial display of the imported data.

After the omissions were corrected, the file was successfully imported into PharmGKB. We subsequently submitted a query to PharmGKB to view some of the polymorphism data for exon 4 of HNMT (Figure 5). This confirmed that the data had been successfully imported. While PharmGKB reports are tabular, the data is stored in the ontology as a set of interlinked instances; the links are automatically created while parsing the XML document.

Our ontology evolved after we began collecting data; occasionally, a new field was added, or the constraints on a data value type changed. When this happened, we generated a new XML schema after modifying the ontology and published it on the PharmGKB web site. To date, this approach to automating the updating of the XML schema interface has been successful and appears to be scaling well.

4 Discussion

Pharmacogenetics spans a broad range of information which must be synthesized in order to find possible connections between genotype and drug response. Ontologies are useful for modelling complex domains such as pharmacogenetics, and their benefits in bioinformatics have been previously described.[14,15] Most of the existing biology data resources are databases rather than knowledge bases: they describe miscellaneous objects according to the database schema, but no representation of the general concepts and their relationships is given.[27] Because of the large number of diverse concepts and relationships among them in pharmacogenetics, the PharmGKB data model is based on an ontology.[16]

Our work addresses the problem of creating a robust interface between an ontology and data acquisition for that ontology, such that when the ontology changes, the process of updating the interface can be automated. Our approach involves (1) using an XML schema to define the mapping from data acquisition to the ontology, (2) encoding the constraints that define the XML schema directly into the ontology, and (3) designing the XML schema to have related data are grouped together so that users submitting data can map their relational data directly into an XML document.

The approach taken to data integration in databases has been to either create a data warehouse[28] or create mappings between the sources.[29] Static mappings applied to ontologies would be difficult to maintain as the ontology changes. In our method, we establish a "common data model," specified in XML schema, shared by the ontology and an external relational data source (the study centers). Common data models have been used previously with relational databases.[21] We chose XML because it is self-describing, flexible, it can closely reflect ontology models, and it can facilitate semantic interoperability.[30]

Data acquisition for an ontology is usually done by a user who creates instances and fills in their slot values directly.[14,15] Collecting data as instances makes sense if one has an intimate understanding of the ontology and the user's model of the data is instance-based. But scientists who collect experimental data usually think in terms of tables, not in terms of instances in an ontology. When submitting data on PCR assays, the primers are part of the information about the assay; in the PharmGKB ontology, the primers are separate data objects. It is simpler for the user to submit data about primers and PCR assays together, rather than submitting primer information before sending the other data about the PCR assay.

Our solution is to provide an XML schema interface to the ontology that maps directly to the experimental data being collected. Our XML schema nests elements from classes having related information beneath the main submission class. For example, for PCR assays, the elements related to primer submissions are nested beneath those for PCR assay submissions. In this way, the user has a submission interface to PharmGKB that looks relational (Figure 4) while preserving the information required to store the data in a rich hierarchical ontology. We are not aware of a similar approach taken for integrating ontologies with external information.

The benefit of our method is that we can automate the process of updating our interface to data acquisition when the ontology changes—we simply update the XML schema. Because the XML schema is defined from metadata in the ontology, changes to the ontology can be immediately ported into a new version of the XML schema. We are also developing software to make this happen automatically. The user submitting data will still have to update the mappings from their data to the new XML schema, but the XML schema interface resembles a tabular representation that is closer to their own than the ontology.

Our evaluation to date is preliminary. We have shown our approach is feasible and has been successful with real data from one of the study centers. We plan to perform a more complete evaluation of our methodology, a task that will be possible as more study centers begin submitting data to PharmGKB.

A limitation of our method is that changes to XML schemas are generally not backward compatible with older XML documents that were created according to a previous version of the XML schema. This means that older XML documents that have previously been processed cannot be re-processed under the new schema. In addition, our method requires all users to keep current with the latest version of the XML schema. These limitations are typical of any system that declares a standard interface between two different components. However, the benefits of having a standard interface generally outweigh these limitations. In particular, the benefit of being able to integrate outside information in PharmGKB is vital to the project. Furthermore, a new version of the XML schema is automatically defined as the ontology changes, so the effort of maintaining a current interface is much less than the work that would be involved in manually establishing new mappings between the data and the ontology as the ontology changes.

In conclusion, we have developed a method for integrating an ontology of pharmacogenetics with data input from external sources. Our method allows us to preserve a relational view of the data in creating our interface, and it uses XML that keeps the data in a clear, human-readable format. Our approach appears promising with respect to being able to preserve the link between the ontology and external sources even as the ontology evolves and changes. We will use this method for integrating PharmGKB with other resources, and our methods could be applicable to data integration for ontologies in other domains.

Acknowledgments

The PharmGKB is financially supported by grants from the National Institute of General Medical Sciences (NIGMS), Human Genome Research Institute (NHGRI) and National Library of Medicine (NLM) within the National Institutes of Health (NIH) and the Pharmacogenetics Research Network and Stanford University's Children's Health Initiative. This work is supported by the NIH/NIGMS Pharmacogenetics Research Network and Database (U01GM61374). The authors also wish to thank Katrina Easton for her assistance in implementing this work.

References

1. The SNP Consortium Ltd., (2000). Available at http://snp.cshl.org/news/user_survey.pdf.
2. P.O. Brown & D. Botstein, *Nature Genetics* **21**, 33-7 (1999).
3. N. Williams, *Science* **275**, 301-2 (1997).
4. S.T. Sherry et al., *Nucleic Acids Research* **29**, 308-11 (2001).
5. M.P. Skupski et al., *Nucleic Acids Research* **27**, 35-8 (1999).
6. S.I. Letovsky et al., *Nucleic Acids Research* **26**, 94-9 (1998).
7. D.A. Benson et al., *Nucleic Acids Research* **28**, 15-8 (2000).
8. D. Jacobson & A. Anagnostopoulos, *Trends in Genetics* **12**, 117-118 (1996).
9. A. Hamosh et al., *Human Mutation* **15**, 57-61 (2000).
10. C. Harger et al., *Nucleic Acids Research* **28**, 31-32 (2000).
11. National Institute of General Medical Sciences, National Institutes of Health, (2001). Available at http://www.nigms.nih.gov/funding/pharmacogenetics.html.
12. R. Stevens, C.A. Goble & S. Bechhofer, *Briefings in Bioinformatics* **1**, 398-414 (2000).
13. P.G. Baker et al., *Bioinformatics* **15**, 510-20 (1999).
14. P.D. Karp et al., *Nucleic Acids Research* **26**, 50-3 (1998).
15. R.B. Altman et al., *IEEE Intelligent Systems & Their Applications* **14**, 68-76 (1999).
16. T.E. Klein et al., *The Pharmacogenomics Journal* (2001 -- in press).
17. S. Schulze-Kremer, "Integrating and Exploiting Large-Scale, Heterogeneous and Autonomous Databases with an Ontology for Molecular Biology" in *Molecular Bioinformatics, Sequence Analysis - The Human Genome Project* (ed. H. Lim) 43-56 (Shaker Verlag, Aachen, 1997).
18. T. Bray, J. Paoli & C.M. Sperberg-McQueen, (1998). Available at http://www.w3c.org/TR/1998/REC-xml-19980210.html.
19. S. Staab et al., AIFB, University of Karlsruhe, Technical Report 401 (2000).
20. S. Bowers & L. Delcambre, ECDL 2000 Workshop on the Semantic Web, 2000). Available at http://www.ics.forth.gr/proj/isst/SemWeb/proceedings/session1-1/html_version/.

21. D. Gardner et al., *Journal of the American Medical Informatics Association* **8**, 17-33 (2001).
22. F. Achard, G. Vaysseix & E. Barillot, *Bioinformatics* **17**, 115-125 (2001).
23. G.C. Xie et al., *Bioinformatics* **16**, 288-289 (2000).
24. P. Tarczy-Hornoch et al., *Journal of the American Medical Informatics Association* **7**, 267-76 (2000).
25. XML Schema Working Group, (2001). Available at www.w3.org/TR/xmlschema-1/; www.w3.org/TR/xmlschema-2/.
26. M.A. Musen et al., Proceedings of the Conference on Intelligent Information Processing (IIP 2000) of the International Federation for Information Processing World Computer Congress (WCC 2000), Beijing (2000).
27. C.D. Hafner & N. Fridman, *Proceedings of the International Conference on Intelligent Systems for Molecular Biology; ISMB* **4**, 78-87 (1996).
28. O. Ritter et al., *Computers & Biomedical Research* **27**, 97-115 (1994).
29. T. Etzold, A. Ulyanov & P. Argos, *Methods in Enzymology* **266**, 114-28 (1996).
30. S. Decker et al., *IEEE Internet Computing* **4**, 63-74 (2000).