Databases and ontologies

Chemical effects in biological systems (CEBS) object model for toxicology data, SysTox-OM: design and application

Sandhya Xirasagar1,* , Scott F. Gustafson1, Cheng-Cheng Huang1, Qinyan Pan1, Jennifer Fostel3, Paul Boyer1, B. Alex Merrick2, Kenneth B. Tomer2, Denny D. Chan1, Kenneth J. Yost III1, Danielle Choi4, Nianqing Xiao1, Stanley Stasiewicz2, Pierre Bushel2 and Michael D. Waters2

1Science Applications International Corporation, 20201 Century Boulevard, 3rd Floor, Germantown, MD 20874, USA, 2NIEHS, National Center for Toxicogenomics, 3Lockheed Martin Information Technologies, PO Box 12233 and 4Research Triangle Institute, PO Box 12194, Research Triangle Park, NC 27709, USA

Received on October 6, 2005; revised on December 20, 2005; accepted on January 5, 2006
Advance Access publication January 12, 2006
Associate Editor: Charlie Hodgman

ABSTRACT

Motivation: The CEBS data repository is being developed to promote a systems biology approach to understand the biological effects of environmental stressors. CEBS will house data from multiple gene expression platforms (transcriptomics), protein expression and proteome–protein interaction (proteomics), and changes in low molecular weight metabolite levels (metabolomics) aligned by their detailed toxicological context. The system will accommodate extensive complex querying in a user-friendly manner. CEBS will store toxicological contexts including the study design details, treatment protocols, animal characteristics and conventional toxicological endpoints such as histopathology findings and clinical chemistry measures. All of these data types can be integrated in a seamless fashion to enable data query and analysis in a biologically meaningful manner.

Results: An object model, the SysBio-OM (Xirasagar et al., 2004) has been designed to facilitate the integration of microarray gene expression, proteomics and metabolomics data in the CEBS database system. We now report SysTox-OM as an open source systems toxicology model designed to integrate toxicological context into gene expression experiments. The SysTox-OM model is comprehensive and leverages other open source efforts, namely, the Standard for Exchange of Nonclinical Data (http://www.cdisc.org/models/sendv2/index.html) which is a data standard for capturing toxicological information for animal studies and Clinical Data Interchange Standards Consortium (http://www.cdisc.org/models/sdtm/index.html) that serves as a standard for the exchange of clinical data. Such standardization increases the accuracy of data mining, interpretation and exchange. The open source SysTox-OM model, which can be implemented on various software platforms, is presented here.

Availability: A universal modeling language (UML) depiction of the entire SysTox-OM is available at http://cebs.niehs.nih.gov and the Rational Rose object model package is distributed under an open source license that permits unrestricted academic and commercial use and is available at http://cebs.niehs.nih.gov/cebsdownloads. Currently, the public toxicological data in CEBS can be queried via a web application based on the SysTox-OM at http://cebs.niehs.nih.gov.

INTRODUCTION

Biological systems respond to environmental stressors in complex ways that comprise both generic responses as well as those that are specific to the agent and are thus more directly related to its mechanism of toxicity. High-throughput methods for characterizing gene expression, particularly the ‘omics platforms, are poised to revolutionize biology and comprehensively define biochemical mechanisms of injury and disease resulting in new biomarkers (Amin et al., 2002; Merrick and Tomer, 2003; Waters and Fostel, 2004; Gant and Zhong, 2005; Lord, 2004; Sawada et al., 2005). Time- and dose-dependent ‘omics data will better distinguish varying phases of biologic response to test agents such as pharmacologic, toxic and adaptive reactions. The analysis of ‘omics data including transcriptomics, proteomics and metabolomics profiles in the context of conventional measures of adverse reaction (clinical or histopathological phenotypes) is expected to spur development of new biomarkers of toxicity, distinct mechanisms of toxic injury in response to different classes of toxicants, signature pathways of toxicity and genotypic effects of toxicity. Such studies require collective analyses of disparate types of gene expression data generated under different conditions, using several agents with varying levels of exposure. Recently, increasing availability of computational resources (such as hardware and memory) has led to the usage of increasingly complex computational tools to perform integrative mining of these diverse types of data. Furthermore, these resources are also being used to address issues relating to discrimination of
treatment-specific responses from effects owing to confounding factors. However, currently toxicological data in most cases are stored in a textual form with very limited use of controlled vocabularies for annotation, rendering it unwieldy for effective automated computational query and analysis with sophisticated analytical tools. Such considerations demonstrate the need to capture study details including the study design, test protocols and the data, comprehensively and accurately in a standardized manner using controlled vocabularies.

The Chemical Effects in Biological Systems (CEBS) toxicogenomics data repository is being designed to enable users to perform integrative mining of conventional toxicology and 'omics data. Hence, CEBS has been designed to capture all the details of a toxicogenomics experiment workflow in a queryable manner using controlled vocabularies for annotations. The details specific to 'omics workflows together with the description of the SysBio-OM model have been designed to capture such information (Xirasagar et al., 2004). The study design and conventional toxicology test data details pertaining to a toxicogenomics experiment are described below.

**Study design**

A toxicogenomics study, similar to any other biological study, begins with the specification of a 'study design' which involves choosing the appropriate subject or experimental system and designing a set of conditions under which specific tests will be carried out to address the specific objectives of the study as unambiguously as possible. Most toxicogenomics studies use the concept of groups to associate biological replicates that are all treated in a similar manner to gauge the reproducibility of experiments. Events such as treatment, to specifically address the objectives of the investigator, auxiliary events such as diet, animal husbandry, culture conditions, method of sacrifice, etc. and the tests are usually scheduled for groups. Events are performed according to protocols with specific parameter values which need to be documented. The time (actual or relative to the study beginning or to another event) for each event is recorded, so that one can determine how the different events relate to one another in the time dimension.

**Biological tests and findings**

During or following a treatment, results are obtained from tests that are designed to gauge the responses of the subjects. Tests can be conducted on subjects (body weights) or on samples (also termed specimens) obtained from subjects (e.g. clinical chemistry measures of enzyme activities and other small molecules in blood). The relationships between the subjects and their samples and the subjects/samples and tests need to be recorded. Specifics of the sample preparation and sample storage also need to be recorded since these procedures can affect the data.

'Biological tests' are performed using specific protocols that result in test data which can be graphical (e.g. images), or simple textual (e.g. body/organ weight, clinical chemistry measures), or complex data such as histopathology, which is hierarchically structured. Data derived from analyses (e.g. pathology finding) must be linked to the source data (e.g. image) and the analysis methods used for the analyses.

The toxicological context for toxicogenomics experiments as described above needs to be captured accurately and comprehensively so as to fully represent the state of the target system. Additionally, standardization and open systems are essential for meaningful exchanges of data between repositories so as to facilitate large-scale collective data analyses. Recent standardization efforts include the Minimal Information About a Microarray Experiment (MIAME)/Tox guideline by the Toxicogenomics Working Group under the Microarray Gene Expression Database (MGED) society; http://www.mged.org/Workgroups/rsbi/MIAME-Tox-Checklist.doc, the Standard for Exchange of Nonclinical Data (SEND; http://www.cdisc.org/models/send/v2/index.html) which is a data standard for capturing toxicological information for animal studies and Clinical Data Interchange Standards Consortium (CDISC; http://www.cdisc.org/models/sdtm/index.html) have been started. The MGED Reporting Structure for Biological Investigations (RSBI) working group is an international effort to evolve a unified reporting concept fort reporting complex biological investigations employing omics-technologies. The idea is to use an ontologically grounded conceptual framework to enable 'semantic integration' of complex data, data mining and information retrieval (http://www.mged.org/Workgroups/rsbi/rsbi.html; Sansone et al., 2004). Another public effort in the toxicology community has led to the development of ToxML which is an XML (Extensible Markup Language; http://www.w3.org/TR/xml11/) schema standard for chemical toxicity data (http://www.epa.gov/nheerl/dsstox/).

Other efforts include the Distributed Structure-Searchable Toxicity (DSSTox) Public Database Network, the members of which support the Standard Data Format for representing multiple chemical structure records and associated data fields.

**RESULTS**

An open source model does not currently exist to capture complex hierarchical toxicological data to facilitate programmatic storage and complex querying. The MAGE-OM developed by the MGED society (http://www.mged.org/) was designed to capture microarray experiments including experimental conditions under which a sample for hybridization was obtained, the experimental protocols and the resulting data. The SysBio-OM (Xirasagar et al., 2004; http://cebs.niehs.nih.gov) was developed earlier as an extension of MAGE-OM to represent the other 'omics data namely proteomics and metabonomics. However, the preceding models were not designed to capture the true complexity of a toxicological study design and data over time. For example, a user may be interested not only in knowing exposures to environmental agents which result in a similar phenotype (e.g. effects on organs/cell types, pathology or 'omics profile), but would also like to know the manner in which the agent was delivered or the diet/animal husbandry conditions of the animal at the time of treatment, and how these factors affected the resulting phenotype. Such questions emphasize the need to monitor changes in expression of genes and proteins and metabolite levels in the context of conventional toxicology endpoints (e.g. clinical chemistry and histopathology) the study design, describing the detailed toxicological context. (Waters and Fostel, 2004; Lord, 2004).

The capture of the toxicological context for omics experiments required a choice of either extending the CEBS SysBio-OM single model or designing a decoupled system. Several considerations summarized below led us to opt for the latter modular system approach, whereby the CEBS Systems Toxicology object model
(SysTox-OM) has encapsulated the complex toxicological context of a toxicogenomics experiment. The integration with the 'omics data in the SysBio-OM is accomplished via the subjects and/or samples, cross-referenced between the two object models. In such a system as CEBS which is envisioned to host comprehensive and diverse data, a modular approach would reduce complexity and maintenance efforts as well as enable rapid development, troubleshooting and extension of each module. Each active application can be implemented independently and be unaware of the complex details of the other applications. As the systems grow in size and processing needs, it is relatively easy to horizontally scale operations to accommodate increased user demand, since each of the applications can be housed independently. Modular systems do, however, require more rigorous testing of each individual system to ensure accurate functionality and performance. Below, we describe the SysTox model, a fine grained and structured representation of the toxicological context for the 'omics samples that will enable in-depth querying among experiments.

The SysTox-OM includes elements of MIAME and MIAME/Tox guideline (Brazma et al., 2001; http://www.mged.org/Workgroups/rsbI/MIAME-ToxChecklist.doc), the SEND (http://www.cdisc.org/models/send/v2/index.html) and CDISC (http://www.cdisc.org/models/sdm/index.html). A CEBS Data Dictionary (CEBS-DD; Hostel et al., 2005) has been recently developed to map vocabulary from multiple sources for data capture from other repositories, ensure accurate annotation of all toxicological experiments and facilitate generation of output files according to the above consortia criteria. Incorporation of standard vocabulary and ontologies should improve the quality and utility of data and make use of alternative data/annotation sources.

Description of the SysTox-OM

The SysTox-OM is designed to foster the efficient capture, storage, retrieval and exchange of study design and conventional toxicology information such as clinical chemistry, hematology, clinical observations and histopathology. A Universal Modeling language (UML) (http://www.rational.com; Rumbaugh et al., 1999) depiction of the entire object model is available at http://cebs.niehs.nih.gov/.

Elements and events are represented by classes (shown as boxes), that have attributes and may have varied relationships such as aggregation, inheritance, directed or bidirectional association (shown as different types of lines and explained in the legend of Fig. 1) to one another. Toxicological annotation is organized into 10 different domains, termed packages that contain classes that share a common purpose and are shown in Table 1.

In the following sections, a description of SysTox-OM is presented. Since a comprehensive description of all the classes in the model is beyond the scope of this publication, we selected the main aspects of the model to describe its utility and have provided additional material as Supplementary information at http://cebs.niehs.nih.gov/

SysTox-OM, similar to other object models such as MAGE-OM and the SysBio-OM, is a data-centric model and has incorporated concepts from MAGE-OM where possible. There are five abstract classes in the SysTox-OM from which all the other classes in the model are derived: SysToxObject, Describable, Identifiable, Parameterizable and ParameterizableApplication. Abstract classes provide common behavior across a set of subclasses, but by themselves cannot be instantiated. All other concrete classes (those that can be instantiated) are extended from one of these superclasses and therefore inherit the properties of their respective superclasses, while having additional properties of their own.

In order to capture specific relationships such as 'composed of', 'belongs to', 'sibling of', 'employed by', 'before', 'causes' between objects, an interface Relationable was modeled. Objects depicting organisms, parties, events, findings, etc. are associated with one another via realization classes of the interface Relationable, providing the means to capture the relationships between these objects. Associations between the realization classes of the interface Relationable and class SysToxOntology (see below) enforce the usage of controlled vocabulary to describe the relationships, enabling effective querying by relationships. We describe the packages and the classes below to accurately represent a toxicological experiment.

CommonHierarchy, Contact, and Measurement Common-Hierarchy, Contact and Measurement packages are described in detail in the Supplementary Materials. The CommonHierarchy package is based mainly on the MAGE-OM, and includes classes to support requirements shared by the classes described below, specifically the ability to specify simple annotations, bibliography, free-text descriptions and unique identifiers that have unambiguous reference within the scope. The Contact package captures all of the information pertinent to users and their roles in different contexts. The Measurement package enables the annotation of measurements and the respective units for measurements. Class Measurement is extended to represent Numerical, Categorical or Boolean values.

The association between classes Categorical and SysToxOntology ensures that the categorical values (e.g. severe, mild) are derived from controlled vocabularies, which may be ontologies. Class Numerical points to class Unit, which is extended to support the different types of units. Controlled vocabularies will be used for units.

SysToxOntology. Annotations of objects may be controlled (or not) depending on the type of annotation. For example, the title for a study is not a controlled vocabulary. In cases where the annotation does not need to be controlled, explicit annotations are provided in the model via attributes. Controlled vocabulary is necessary to enforce consistency for specific terms with specific meanings enabling effective storing and querying of the data. Controlled vocabularies to annotate the objects in the model are provided via domain-specific vocabulary extensions of classes in the SysToxOntology package. Separation of the base model from the annotations allows the modular development of domain specific annotations independent of the base model thus allowing such efforts to proceed independently of one another based on the needs of the system. Controlled vocabularies required to describe objects are largely provided via relationships of the objects to class SysToxOntology or its extensions.

Explicit relationships or ontologies between controlled vocabulary terms allow the data to be stored non-redundantly while enabling powerful queries. For example, a user does not need to specify that the animal used was a rat and a mammal if the relationship between the term, mammal and rat, in the ontology is stored in the system. A query for terms, mammal, would imply that the animal used was a rat and a mammal if the relationship between the term, mammal and rat, in the ontology is stored in the system. A query for terms, mammal, would implicitly include rat related queries. Such relationships are specified in the model via associations between the class SysToxOntology and the realization classes of the interface Relationable (see above).
Fig. 1. Universal Modeling language (UML) model of representative elements of the ‘Study Package’ in the SysTox-OM. The core element of the class diagram is the class. Classes are used to represent entities or objects. Lines connecting classes denote associations between classes with the type of line indicating the type of relationship such as generalization and aggregation. The directionality of the relationship can be indicated by an arrow (→) showing the flow of information between classes and is driven by the manner in which the data can be most usefully accessed. The class at the end away from the arrow has access to information provided by the class at the end with the arrow. The cardinality specifying the number of expected instances of the accessed class in a relationship is denoted either on the left of (for vertical) or above (for horizontal) the line representing the relationship. Bidirectional relationships, in which both associated classes have access to each other’s information, are indicated by straight lines with no arrows. Generalization or inheritance or an ‘is a’ relationship (→) is indicated by a line with a triangle at one end and refers to a relationship between two classes where one class, the subclass, is a specialized version of another, the superclass (the end with the triangle). In other words, the subclass inherits all of the superclass attributes, and the subclass also has one or more attributes specific to it. For example, mouse is a subclass of the superclass mammals since all mammals share some common features, but mice have certain attributes not present in other mammals. Aggregation (→) is denoted by connecting two classes with a line ending in a rhombus. The class towards the rhombus end represents an aggregation or composition of one or more classes at the other end of the line. For example, a Project includes one or more Study(s). Some elements are repeated in the packages to depict the connectivity with other elements. Classes not belonging to the package being depicted will display the name of the package that the class belongs to in parenthesis below the class name. A more detailed description of the UML and its components are available in Rumbaugh et al. (1999), and at http://www106.ibm.com/developerworks/rational/library/769.html

See Supplementary figures for a depiction of the current extensions in SysToxOntology.

Study. The Study package provides access to the high level information about a study via classes and the relationships between the classes (Fig. 1). A project or investigation is used to define a set of related studies with each study having its own study design. The species, strain, age, sex, disease model, cell-type (for an in vitro experimental system) and other descriptions pertaining to the subject are factors included in the study design definition. The Study class provides information about the title, objectives of a study and if it was a GLP (Good Laboratory Procedure) study. Study’s relationship to the class. Project, enables the inference of the project (or investigation) to which it belongs. Study is associated with StudyDesign, which captures the high-level study design information by its associations to the StudySchedule (schedule of planned events such as, treatment, tests, sacrifice), Stressor (the stressor being investigated) and Subject (the experimental systems such as organism, cell culture, etc. in which the response to the stressor is being studied). Study-time related issues like duration of the study, begin and end dates of the study and study contacts are also captured by objects in this package via relationships between the Study class and TimeValue and PartyRole classes, respectively. During the course of a study, certain deviations with regard to the original study design may occur. Hence, in addition to the planned events, the deviations also need to be recorded. A Study can access deviations in the study schedule via its relationship to StudyExecution which has a relationship to the actual event occurrences.

Subject. The Subject package contains base classes that provide annotations for the experimental system under study (Fig. 2). Associations between the class Subject and classes derived from SysToxOntology provide experimental annotations including characteristics (sex, taxonomy, disease, age, cell line, etc.). Subjects can be individuals or groups represented by classes Individual and Group extended from Subject, respectively. Relationships among
ultimately be treated identically (especially in case of clinical sub-
planned for these groups. However, individual subjects may not
exposed to the same treatment/testing regimen, the study design
for any hardware or software used for
have deviated for each of the protocol parameters, as well as setting
software and can have a list of parameters for which default values
can be specified. Individual parameter values can change between
individual uses of the protocol. In order to specify the values that
have deviated for each of the protocol parameters, as well as setting
the protocol parameters for any hardware or software used for
that particular instance, class ProtocolApplication was modeled. ProtocolApplication enables non-redundant storage of protocol
information. If there are minor deviations from a protocol,
then the protocol is only stored once. Deviations in protocol are
recorded for each use of that protocol. Several extensions to the
base class Protocol (e.g. DosingProtocol) and Parameter
(e.g. DosingParameter) enable the specific description of specific
types of events (see below under Event).

**Protocol.** The Protocol package captures protocol descriptions
(e.g. treatment protocol) to perform events (e.g. treatment) (Fig. 3
and Supplementary Materials). Protocols can use hardware and/or
software and can have a list of parameters for which default values
are defined. Individual parameter values can change between
individual uses of the protocol. In order to specify the values that
devoided for each of the protocol parameters, as well as setting
the protocol parameters for any hardware or software used for
that particular instance, class ProtocolApplication was modeled. ProtocolApplication enables non-redundant storage of protocol
information. If there are minor deviations from a protocol,
then the protocol is only stored once. Deviations in protocol are
recorded for each use of that protocol. Several extensions to the
base class Protocol (e.g. DosingProtocol) and Parameter
(e.g. DosingParameter) enable the specific description of specific
types of events (see below under Event).

**Event.** Classes in the Event package capture the description of
how, when and what event was performed on which subject(s) or
sample(s) (Fig. 3) in a Study. The protocol used to perform the
Event is captured by the association between the classes, Event and
Protocol (Fig. 3). The study designs in a toxicological experiment
often involve testing biological replicates grouped together as treat-
ment groups. Since each individual within the treatment group is
exposed to the same treatment/testing regimen, the study design
specifies the schedule of events performed on the different treatment
groups and the protocols for performing these events. The class,
Event, captures a planned event and can be used to annotate an event
planned for these groups. However, individual subjects may not
ultimately be treated identically (especially in case of clinical sub-
jects) since a test subject may unexpectedly expire before scheduled
sacrifice. Hence, the class EventOccurrence derived from Protocol-
Application captures the actual individual events performed on
individual subjects within groups. EventOccurrence points to
Event enabling the inference of the planned event. The class
StudyExecution that points to all of the EventOccurrences enables determination of planned events that actually occurred.
For instance, relationships between individual events that concern
individual animals can be modeled such that individual points to the
EventOccurrence which points to the Event that has been planned
for that group and the group points to the Event. The Event package
enables the capture of relationships between two or more events.
The class Phase derived from the Event is used to represent the term
phase, often used in toxicology to annotate a time period during
which certain events happen (treatment phase, acclimatization
phase, etc.). Extensions of the base class Event enable specific
events such as treatment, test, disposition/demise, sample preparation,
acclimatization of animal prior to treatment, randomization
process to select subjects for the study, etc. to be defined. The time
domain annotation is one of the critical and complex components of
the description of a toxicological experiment and hence the class
TimeValue, a subclass of Measurement was created to annotate the
time domain. Events happen at specific times and can occur over
time (duration). The time of occurrence of an event can be annotated
as actual clock time and/or time elapsed relative to the study begin
and/or time elapsed relative to when other events were performed
(e.g. dosing of animal performed on 10/11/2005 or on day 1 of study
or 4 h after feeding). Relative time of one event with regard to
another can be very useful in cases where event outcomes are
interdependent (i.e. duration between the treatment time and obser-
vation time may affect the observation outcome). Capture of Time
contexts for any Event can occur within the study using relation-
ships between these classes and TimeValue.

**Intervention.** The Intervention package captures the information
relevant to the treatment of the animal (Fig. 4). A study of toxico-
logy may include complex protocols for diet, animal husbandry, cell
culture, administration of stressor(s), etc. The Intervention package
designed in the SysTox model provides a comprehensive way to
capture simple treatment protocols (acute, single stressor treat-
ments) as well as complex treatment designs (chronic, multiple
stressor treatments). The Intervention class, derived from Event,
aggregates individual treatments applied to a subject. This concept
is useful to follow the chronology of events where complex treat-
ment regimens have been followed. For multiple stressors admin-
istered to the same animal, if the treatment includes two stressors
not given at the same time or if two different types of stressors are
administered to the animal (for example, UV radiation and a chemi-
cal stressor), the treatments can be treated as separate treatments.
Hence in this case, the ‘starttime’ of intervention is the time when
the first treatment is initiated and the ‘duration’ of the intervention is
the time elapsed between when the first treatment began and when
the last treatment was completed. If two chemicals are part of one
dose then these two chemical stressors can be used as part of the
same treatment.

**Findings.** The Findings package captures the tests and test data in
relation to the subject/sample on which the tests were conducted
(Fig. 5). This package is described under the two subsections below.

**Sample preparation.** Sample annotations must be captured to
include preparation protocols and characteristics, since both can
significantly contribute to the observed test results, and hence
objects modeling these annotations are part of the Findings package.
The specimen/sample preparation aspect of the Findings Package
includes the classes TestSample and SampleRel. The relationship

---

Table 1. SysTox-OM packages

<table>
<thead>
<tr>
<th>Packages</th>
<th>Functionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>CommonHierarchy</td>
<td>Elements common to all packages</td>
</tr>
<tr>
<td>SysToxOntology</td>
<td>Controlled vocabularies for annotations</td>
</tr>
<tr>
<td>Study</td>
<td>Study design</td>
</tr>
<tr>
<td>Subject</td>
<td>Experimental system(s) (organism/cell culture) that is being studied. Biological materials used for the experiment and description of their creation</td>
</tr>
<tr>
<td>Protocol</td>
<td>Top-level hierarchy and hardware, software protocols</td>
</tr>
<tr>
<td>Event</td>
<td>Top-level hierarchy and relationships for all planned and performed events</td>
</tr>
<tr>
<td>Intervention</td>
<td>Subject treatment, diet/culture conditions, animal husbandry</td>
</tr>
<tr>
<td>Findings</td>
<td>Sample preparation, tests and test data</td>
</tr>
<tr>
<td>Contact</td>
<td>Contact and role information</td>
</tr>
<tr>
<td>Measurement</td>
<td>Numerical, categorical and boolean measurements and units</td>
</tr>
</tbody>
</table>
Fig. 2. UML model of representative elements of the ‘Subject Package’ in the SysTox-OM. The UML model symbols are explained in detail in Figure 1 legend.

Fig. 3. UML model of representative elements of the ‘Protocol Package’ and the ‘Event Package’ in the SysTox-OM. The UML model symbols are explained in detail in Figure 1 legend.
between Subject and TestSample via SampleRel (a realization of the interface Relationable) enables the determination of the source subject/sample to sample relationships. Many relationships can be represented between samples and their sources: the relationship between samples and the subjects from which the samples were derived, samples and the parent samples from which they were derived, pooled samples and the samples that compose them. The class SampleRel has a relationship to the EventOccurrence to capture the time/protocol annotations for sample preparation/storage.

Tests and Test Results. Test is a subclass of event and points to the subject or sample on which it is performed. Class TestProtocol captures how the test was performed. Performing a test on a subject/sample can result in data which are represented by simple result such as clinical chemistry result or a complex result such as a histopathology result. We have therefore extended the class TestData to represent each of these types of findings, Graphical-Output (graphic result), SimpleResult (simple results) and ComplexResult (complex results). Complex results may display a hierarchical structure and include the existence of relationships between the results.

Thus the SysTox-OM has been built by modeling the components and relationships of a toxicological experiment in a modular fashion and has allowed us to successfully capture and browse toxicological data. Below, we briefly describe our implementation of the SysTox-OM.

Implementation of the SysTox-OM

The implementation of the SysTox-OM is based on several open source technologies. An automatic code generator, AXgen (http://axgen.sourceforge.net/) created the Java™ classes,¹ the database schema and the object relational mapping document. Since the database schema was created automatically from the SysTox-OM model, the UML available for download differs slightly from the model since some object names were adjusted for compliance with the Oracle naming convention for database objects. A modified version of AXgen utilized the SysTox-OM XMI file in conjunction with the templates created to generate the classes, the database schema and the mapping between the Java classes and the database objects. The Struts framework (http://jakarta.apache.org/struts/) implemented the web application for the SysTox-OM. Struts has a flexible control layer based on the standard Model-View-Controller (MVC; http://java.sun.com/blueprints/guidelines/designing_enterprise_applications_2e/web-tier/web-tier5.html) design paradigm. Struts provides an extensible environment for custom extension and plug-ins. The web browser application uses several standard tag libraries including Display tag and JSTL to encapsulate functionalities, common web applications and hasten code development and testing effort. The toxicological application server encapsulates all the business for manipulating SysTox-OM objects. The various capabilities for accessing these objects are exposed as services to be manipulated by the user interface or other applications. Using this Service-Oriented Architecture, we can easily integrate SysTox-OM with SysBio-OM (Xirasagar et al., 2004). This architecture also allows the future exposure of the SysTox services to applications outside of CEBS.

¹Java is a trademark of Sun Microsystems Inc. in the United States and/or other countries.
if desired. The toxicological application server is placed in a J2EE container, currently JBoss® (http://www.jboss.com).2

Data validation is an important component of a system that integrates a variety of complex data types. We have performed extensive validation at several levels including the object, data storage and user application levels (see below). Additionally, since the data are complex, a curation process is necessary to ensure that the submitted data are accurate. We have instituted measures at several levels to ensure the data integrity and quality including pre- and post-storing of the data in the database. The CEBS security system was designed outside of the domain-specific object models and will control the visibility (private or public) of all the data in CEBS. This system will be used to ensure that only studies designated as curated by a CEBS administrator will be available as public.

**Toxicological data browser**

We have also designed an interface to browse the toxicological data based upon the study design and toxicological end-point queries (http://cebs.niehs.nih.gov; see Supplementary Materials). Currently, the query terms are limited to the stressor name and the clinical chemistry and histopathology data. Users can query by the stressor to obtain studies using this stressor in the study design with more complicated queries and data visualization planned for the future.

**DISCUSSION**

The CEBS SysTox-OM provides a framework for toxicological context of any study data. The toxicological context for the ‘omics samples in SysBio-OM is available via cross-referencing between the subjects or samples in the SysTox-OM and the samples used in the ‘omics experiments. Using these links, one can integrate the study design information, including treatment information and biological characteristics, as well as the conventional toxicological end-point data with the ‘omics data. Using the SysTox model in conjunction with the SysBio-OM, ‘omics studies can be compared temporally and in a dose-specific manner across multiple studies with phenotypic anchors or treatment conditions as the pivots to facilitate cross-study/platform analyses. Analyses of well-annotated toxicogenomics datasets enable discrimination between effects of

---

Fig. 5. UML model of representative elements of the ‘Findings Package’ in the SysTox-OM. The UML model symbols are explained in detail in Figure 1 legend.

2JBoss is a registered trademark of JBoss, Inc. in the United States and/or other countries.
study design factor(s) from those of confounding factor(s). A web-based application is currently under development to integrate transcriptomics, proteomics and the toxicological data using the subject/sample identifiers as the links between these data. This application will appropriately filter/analyze the microarray or proteomics data in the context of the study design information such as treatment parameters, or subject characters, or conventional toxicological phenotypes such as clinical chemistry or histopathology data. The proposed web-interface incorporates a two-tier mode of search, wherein a basic search page would include commonly used search terms while a more advanced search page would include additional less commonly used terms. The potentially large number of search terms will be organized under categories [e.g. study, treatment types (chemical, genetic, physical etc.), animal care, subject characteristics, test types, sample types etc.] to simplify the search process.

Controlled vocabularies are a critical tool for filtering and analyzing annotations. The SysTox object model supports the use of controlled vocabulary and ontologies. Available controlled vocabularies focus upon preclinical data annotations in CEBS. Such controlled vocabularies will be extended and organized into ontologies that leverage other open source controlled vocabularies and ontologies (http://obo.sourceforge.net/) Appropriate extensions include the vocabularies for pharmacokinetics and pharmacodynamics (i.e. Duclos-Cartolano et al., 2003). Furthermore, equivalent concepts existing in ontologies from different resources can be mapped via Universal Resource Identifiers in order to allow users to leverage alternative ontologies. SysTox object model extensibility can adapt to newer experimental platform and modules for independent development of annotation. Such flexibility enables ad hoc software development to continually occur without breaking existing modules.

Use of standard vocabularies and ontologies promotes efficient storage and exchange of annotated data for access to multiple annotation resources. For instance, the Tox-MIAMExpress, an annotation submission tool to ArrayExpress, a public database at EBI based on MAGE-OM (Mattes et al., 2004; http://www.ebi.ac.uk/tox-miamexpress) integrates gene expression and toxicological data via MGED Ontology’s descriptive power. Data elements designed within open source formats like SEND and CDISC standards can be captured within the SysTox model. The CEBS-DD (Fostel et al., 2005) will enable concept mapping between these repositories and CEBS, for data capture from multiple sources. We are collaborating with other large data repositories as well as individual submitters in other institutions to encourage data submission to CEBS. For instance, we are currently collaborating with the EBI to exchange toxicogenomics data captured within ArrayExpress, using a combination of MAGE-ML and other standardized format documents. We are greatly encouraged by policies instituted by several journals that require submitters to make the data available to the public. Sophisticated and domain-specific, mathematical models are being used to simulate, design and interpret experiments in varied biological fields to advance biological understanding. Concerted analyses of ’omics and conventional toxicological data from several replicates derived from different sources can generate highly predictive combinatorial (’omics and conventional toxicology end-points) groups of molecular markers of disease and toxicity. Such integrative analyses can also assist in clarifying the mechanism of action of toxic agents and in deciphering the underlying biological pathways.

The SysTox-OM described here presents an extensible architecture to capture and query virtually any biological experiment and its contextual metadata. We hope that members of the scientific community find the SysTox-OM model to be a useful tool for achieving a greater depth of biological meaning for their experimental toxicogenomics data and for sharing data and technical resources.

ACKNOWLEDGEMENTS

The authors thank Dr Susanna-Assunta Sansone, Harris Johnson, Dr Leping Li and Dr Hong Dang for providing valuable suggestions regarding this manuscript. This project has been funded in whole or in part with Federal funds from the Intramural Research Program of the NIH, NIEHS and Contract No. N01-ES-25495 to Science Applications International Corporation and Contract No. 273-02-C-0027 to Lockheed Martin Information Technologies from NIEHS. Funding to pay the Open Access publication charges was provided by the Intramural Research Program of the NIH, NIEHS and contract no. N01-ES-25495 to Sciences Applications International Corporation.

Conflict of Interest: none declared.

REFERENCES


882