ARTICLES

Bringing Chemical Data onto the Semantic Web

K. R. Taylor, R. J. Gledhill, J. W. Essex, and J. G. Frey*

School of Chemistry, University of Southampton, SO17 1BJ United Kingdom

S. W. Harris and D. C. De Roure

Electronics and Computer Science, University of Southampton, SO17 1BJ United Kingdom

Received September 6, 2005

Present chemical data storage methodologies place many restrictions on the use of the stored data. The absence of sufficient high-quality metadata prevents intelligent computer access to the data without human intervention. This creates barriers to the automation of data mining in activities such as quantitative structure−activity relationship modelling. The application of Semantic Web technologies to chemical data is shown to reduce these limitations. The use of unique identifiers and relationships (represented as uniform resource identifiers, URIs, and resource description framework, RDF) held in a triplestore provides for greater detail and flexibility in the sharing and storage of molecular structures and properties.

1. INTRODUCTION

The comparatively recent rise of the Internet has brought with it unprecedented communication and data exchange between scientific researchers. In parallel with this, the computational power available to the average chemist has increased dramatically in very few years. Other advances in automation and experimentation have led to combinatorial and high-throughput chemistry, and the increased productivity yields enormous quantities of data that present serious difficulties for the existing mechanisms for processing, sharing, and publication. This article demonstrates a new application of Semantic Web technologies to store chemical data in ways that remove many of the obstacles to automated processing and dissemination of the large quantities of richly detailed information that is created by chemists.

The present day state and immediate future of e-science and the Web is discussed in this section, touching on many of the issues encountered by the Comb*e*Chem e-Science project[1] with digitally supported science. Section 2 looks at various Web technologies and their limitations and justifies the use of RDF[2] (resource description framework) to help deal with the problems mentioned above, while Section 3 addresses the issues of describing chemical property data in as complete a manner as possible. The results of creating and using a prototype database for chemical properties are given in Section 4 along with a range of the potential benefits of developing this approach further. Section 5 discusses our investigations into the scalability of RDF-manipulating software, and the final section discusses scope for the system to support computation in a semantic datagrid.

1.1. What is Comb*e*Chem? Combinatorial chemistry is an example of a domain in which new experimental techniques massively accelerate the experimental process by parallelization. The synthesis of new chemical compounds by combinatorial methods provides major opportunities for the generation of large volumes of new chemical knowledge. Given the throughput of data created with combinatorial chemistry, it is not reasonable for every new compound to undergo the same degree of individual analysis and be the subject of a traditional scholarly publication. This introduces a massive bottleneck to dissemination—the majority of the data is left exactly as it came out of the original experiment, one set of fields among many in a matrix of results. Even prior to these high-throughput methods, a significant portion of scientific data remained unpublished, but now the problem is amplified such that a huge amount of data may never be reused unless it is actively sought and harvested by an organization. Commercial companies may not be interested in sharing data, but they, nevertheless, face issues with making the most of their proprietary data.

Computational chemistry is a similarly prolific area of research capable of producing enormous volumes of data. Typically, only the knowledge derived from the data is worthy of publication, but the basic data is, nevertheless, useful for further analysis or verification. These data typically reside in the personal archives of individual researchers, devoid of annotation and reliant on good book-keeping if it should be needed again. In this state, it is also difficult to discover the existence of the data without the aid of the author. There is a strong need for better ways to archive both result and process data.

The need to handle the data deluge is the common characteristic of many of the projects in the e-Science program.[3] The principal drive of the Comb*e*Chem e-Science project is to work with this glut of data to enhance the correlation and prediction of chemical structures and properties by increasing the amount of knowledge available about materials via the synthesis and analysis of large compound libraries.

One of the project objectives is to achieve a complete end-to-end connection between the laboratory bench and the intellectual chemical knowledge that is published as a result

10.1021/ci050378m CCC: $33.50 © 2006 American Chemical Society
Published on Web 01/26/2006

of the investigation, described as "publication at source".[4] The creation of original data is accompanied by data describing the experimental conditions in which it is created, collected with a view to distribution and searching of that data. There then follows a chain of processing such as aggregation of experimental data, selection of a particular data subset, statistical analysis, or modeling and simulation. The handling of these data may include annotation of a diagram or editing of a digital image. All of this generates secondary data, accompanied by the information that describes the process that produced it, and this may be maintained in a variety of distinct datastores. Through the principle of publication at source, all these data are made available for subsequent reuse in support of the scientific process, subject to appropriate access control.

While some calculations may be the application of small-scale statistical models, many of the steps in the process require significant computing resources. The statistical model building in QSAR/QSPR (quantitative structure−property/activity relationship) studies requires access to a large range of diverse chemical information, much of which may be calculated by ab initio quantum codes and molecular dynamics simulations or obtained through experimental means. Both of these latter computational techniques frequently require large computing resources. In both cases, access to a computational Grid infrastructure directly benefits the automation of the calculation workflows by providing the means for easy use of purpose-built resources.

1.2. The Need for Semantics. In 2001, a number of researchers working at the intersection of the Semantic Web, Grid, and software agent research and development communities became conscious of the gap between the aspirations of the e-science vision and the current practice in Grid computing. Concerned that the Grid alone would not meet the e-science requirements, they articulated the potential benefit of applying Semantic Web technologies[5] to Grid infrastructure and applications in the 2001 report "Research Agenda for the Semantic Grid: A Future e-Science Infrastructure".[6,7] The report drew on the Comb*e*Chem scenario as a case study.

The Semantic Web is an initiative of the Worldwide Web Consortium (W3C)

> ...to create a universal medium for the exchange of data. It is envisaged to smoothly interconnect personal information management, enterprise application integration, and the global sharing of commercial, scientific and cultural data. Facilities to put machine-understandable data on the Web are quickly becoming a high priority for organisations, individuals and communities. The Web can reach its full potential only if it becomes a place where data can be shared and processed by automated tools as well as by people. For the Web to scale, tomorrow's programs must be able to share and process data even when these programs have been designed totally independently.
>
> (W3C Semantic Web Activity Statement[8])

While the Grid potentially provides the necessary distributed systems infrastructure, the Semantic Web provides the complementary capability with respect to distributed information. Hence, the combination of the two, a Semantic Grid, enables scientists to answer questions which involve the integration of scientific data and automatic execution of computations, providing important functionality at the scientific applications level. It also facilitates automation within the Grid middleware (software that connects users and other software to the services they require)—helping to discover and compose a variety of Grid resources and services in order to meet the dynamic requirements of multiple Grid applications. Hence, the Semantic Grid is about the use of Semantic Web technologies both on and in the Grid.[9]

Some of the Semantic Web's "added value" comes from accumulating descriptive information (metadata) about the various artifacts and resources. For example, as different stages of the scientific process work with the same referents, perhaps a sample for analysis, a piece of equipment, a chemical compound, a person, a service, or a publication, metadata can be recorded in various stores, in databases or on Web sites. Any kind of content can be enriched by the addition of semantic annotations in this way. This distributed metadata is interlinked by the fact that it describes the same objects, which in turn enables us to ask new kinds of questions which draw on that aggregated knowledge.

Enabling this accumulation of knowledge involves realizing an effective scheme for naming things. The naming problem is facilitated in some areas by existing standards, such as the Life Sciences Identifier,[10] a not yet universal device for biological entities in the life sciences domains, and the InChI (International Chemical Identifier[11]), which in chemistry provides a unique identifier derived from the molecular structure of compounds. These unique identifiers are needed to replace the fundamentally inadequate methods available to us at present. Prior to InChI, chemists were limited to their own arbitrary codes, Chemical Abstracts Reference (CAR) numbers, chemical names both trivial and systematic, or structural descriptions such as SMILES.[12] Arbitrary codes lose all meaning outside of their original setting, and the CAR—a centrally regulated arbitrary code—carries a monetary cost and cannot be supplied for unpublished substances. Chemical names are ambiguous, and systematic names for any remotely complicated species are unwieldy and hard for humans to interpret. SMILES strings are descriptive of the underlying chemistry but lack support for stereochemistry. A collection of these identifiers is needed to locate particular species as no single one is sufficient for all purposes.

1.3. The Power and Importance of Provenance. The origins of data are just as important to understanding as their actual values. The boiling point of a liquid is different if measured in the Alps than if measured at sea level; perhaps a particular person is renowned for the quality of their data. An example of the value of these details can be found in computational chemistry where the accurate geometry of a molecular system strongly influences any calculations performed on it. Selection of the method and basis set dictates the usefulness of an optimization on a case-by-case basis, since what will serve as a good starting point for a molecular surface area calculation will be too crude for a multireference transition state analysis.

A similar scenario arises in laboratory science where a piece of equipment is discovered to be faulty or out of calibration. An inaccurate value can propagate from measurement through analysis and into further derived results;

hence, it is useful to be able to pursue the uses of that value to all ends. In short, knowing who did what, where, and when allows the selection of the best available data for the task at hand and the verification of the validity of that result. Such assessments are usually only possible in an intensely manual way, demanding laborious reference checking, assuming that the values have actually been published. As discussed earlier, this is an untenable approach in high-throughput environments. The addition of carefully chosen metadata greatly expedites this process by allowing software to filter and act on the data.

2. WEB TECHNOLOGIES APPLIED TO SCIENCE

The need to describe data in a platform-independent computer-readable manner is illustrated by the current popularity of XML. XML[13] is a generalized markup language that can be applied to any data (including structured text documents). Many examples now exist of XML being used to annotate data from many fields including geography (GML[14]), biology (CellML[15]), and chemistry (CML[16]) to name but a few. These formats are essentially transfer media for data in those fields, as they are open rather than proprietary formats, and allow exchange of data without a loss of meaning. Typically, they cover most of the scope of the subject area, with the greatest strength on the topics that drove their development. CML provides a workable alternative to the various chemical structure representations and is able to capture detail of molecules at the atomic level. A series of other more specific markup languages also exist, including ThermoML,[17] AniML,[18] SpectroML,[19] and the still apparently developmental NDML and UnitsML. These markup languages focus heavily on recording particular experimental data and are less generic.

Well-implemented XML is a solution to computer-readable data storage, but it does have limitations. Data can benefit from annotation with different metadata languages such as RDF. The Semantic Web requires more than XML descriptions, as XML does not and cannot tell software how to interpret the structures and data it describes.

2.1. What is RDF? RDF[2] is a different type of markup that formalizes the free-form hierarchical structures of XML into triplets of subject, relationship, and object. RDF represents an additional level of detail above XML-based data markup, because it formalizes not only what entities are but also how they relate to one another. While with XML one requires software that understands the particular markup, a generic RDF reasoner can relate one thing to another by the various RDF schemas.

Fundamental to RDF, but less so with XML, is the concept of the URI (uniform resource identifier) whereby one resource is uniquely identified in a global sense. In this way, two documents can contain information about the same resource and can be combined with certitude of their relevance. This capacity to amalgamate dislocated information from multiple domains and sources on demand is the keystone of the Semantic Web. Disparate data linked by a common URI can be drawn together from all over the Web to derive new knowledge using referenced ontologies (discussed below) to determine the meaning of the data.

Each triple in an RDF document typically consists of three URIs: the subject and object, which refer to two entities, and the predicate, which is a URI with a commonly agreed upon meaning, representing the relationship between the subject and the object. For example, to express "isopropyl alcohol has the systematic name propan-2-ol", we could use an RDF triple consisting of the following:

Subject: //molecules/isopropyl_alcohol

Predicate: http://green.chem.soton.ac.uk/rdf/chemschema.rdfs#has-systematic-name

Object: "propan-2-ol"

In a more terse format (N3 triples), this would be expressed as seen in Chart 1, whereas one of the less neat RDF/XML formats would express it as seen in Chart 2.

Chart 1

```
<//molecules/isopropyl\_alcohol> <ch:has-systematic-name> propan-2-ol
```

Chart 2

```
<?xml version="1.0" encoding="utf-8"?>
<rdf:RDF
 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:ch="http://green.chem.soton.ac.uk/rdf/chemschema.rdfs\#">
 <rdf:Description rdf:nodeID="//molecules/isopropyl\_alcohol">
  <ch:has-systematic-name>propan-2-ol</ch:has-systematic-name>
 </rdf:Description>
</rdf:RDF>
```

The Semantic Grid depends on making knowledge explicit and processable by machine, in order to be used in an automated manner. Underlying this is the notion of an ontology. For most practical purposes, an ontology is simply a published, shared conceptualization of an area of content (the extension of terms and the relationships between them), and its primary role is to provide a precise, systematic, and unambiguous means of communication between people and applications. Ontologies provide the basis of metadata. In the example above, the predicate is a URI that would point to an RDF schema document on a web-server that defines "has-systematic-name" as a predicate for IUPAC chemical names. Since ontologies encode relationships between classes of objects, inferences can be drawn between instances; for example, reasoning can be achieved in the OWL (Web Ontology Language[20]) standard using a variety of description logic inference engines. The Web site www.semanticgrid.org contains further information about the Semantic Grid and the Global Grid Forum (GGF) Semantic Grid Research Group.

The primary use for RDF at present is in really simple syndication or RDF site summary[21] (RSS feeds), which convey information across the web to software that renders these summaries for human consumption. Typically, this takes the form of news headlines, but the approach has been applied to summarizing documents containing chemical data in the form of CMLRSS.[22] RDF can do more than this and has a great deal more scope than simply naming authors and summarizing documents. It is a fully capable graph description system, unrestrained by the hierarchical nature of XML. Desired constraints are supplied by the relevant ontologies, but otherwise, it can be used to map relationships between properties, objects, people, and more abstract notions such as classes of objects. Similar relationships can be enforced with XML, through the use of XML schema, but the relationships are not explicit. XML schema languages are

wholly focused on describing syntax rather than information content, and so, XML schema cannot really support the idea of an ontology. They are limited to restrictive rather than an informative role.

2.2. Why RDF? One Technology, Two Benefits. Thus far, all discussion has been focused on issues of transferring data while retaining its meaning and integrity. RDF provides a strong basis for this by unifying relationships and uniquely identified entities into documents. The expression of ontologies in the form of RDF schema (RDFS), OWL, or some other ontology language provides a portable instruction manual for how to interpret any document that invokes them in addition to allowing validity checking.

RDF has uses in the local sense too. So-called triplestores read RDF documents and turn them into graphs that can be explored and used in the same way as older knowledge representation systems (e.g., LISP). Once data is held in a triplestore, it can be searched and manipulated to serve any purpose. Data can be filtered and sorted on the basis of any combination of subjects and relationships.

One can make very convincing arguments for using conventional databasing for local storage, but chemical data is a significant problem for the relational database model. It is multidimensional, and a vast quantity of supplementary information is required to give any particular datum its meaning. The melting point of a particular compound is a common and useful property, but what do we mean when we say that compound X melts at such and such a temperature? It is just a simple number, with units such as Celsius, but the truth of the matter is much more complicated. Compound X is a particular chemical species, but how pure is it? What form has it crystallized in? Did it melt over a range of temperatures? How accurate was the apparatus used to measure the melting point? What if the compound sublimed and never went through the liquid phase? At what pressure was the measurement taken? What if it began to decompose from heating before or while it melted? Perhaps we do not care or do not know, but that does not excuse the datastore from needing to specify these things. Too many data sources gloss over these details, relying on people not requiring them or simply not being aware. The reasons are obvious: this sort of data is intrinsically hard to describe and is easily forgotten or ignored, as well as being more difficult to extract from original literature references.

The normal solution to some of these problems is to supplement the simple number with textual notes. While this may be a quick solution to data input, it makes processing the retrieved numbers much harder, if not impossible. Given that the role of the database is to provide fast and easy access to the data, such a solution is far from satisfactory. To solve these problems with a relational database design results in an exceedingly convoluted and unintuitive data structure. Outside of large commercial companies, this is not an option, as proper database design demands database development professionals, as well as the continued upkeep of such a system, all amounting to a significant expense. The more complex the system, the harder it becomes to alter anything, and it is easily conceivable that the process of scientific research will produce data that it is not possible to store in the database without a complete redesign.

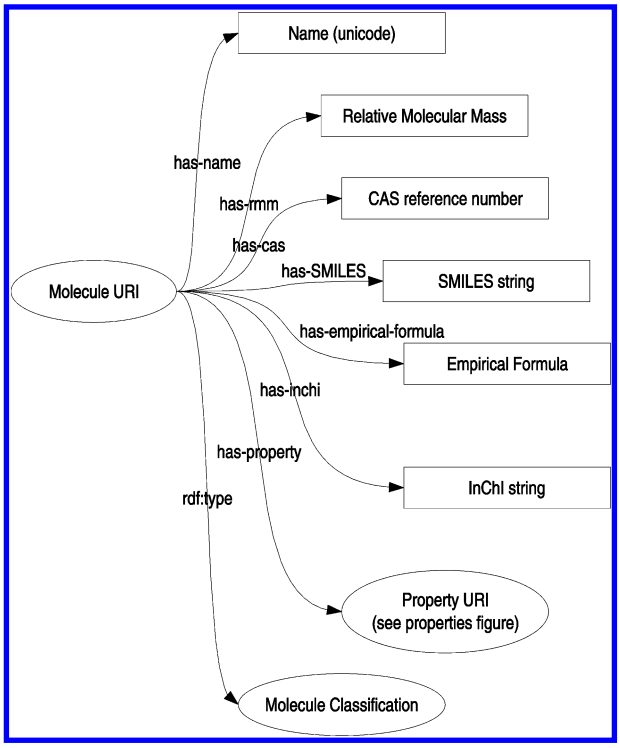

Figure 1. Schema layer 1: state-independent properties.

3. COPING WITH CHEMICAL DATA

Having established the need for semantically rich data and the suitability of RDF for the task, the actual method of describing chemical data must be discussed.

To design the structure of the RDF graph, we identified the identity of most importance, which for this work is the chemical species. It should be noted that this emphasis is a mental tool to assist the design process and understanding of the data. The emphasis could be placed elsewhere since all RDF resources are equally important in software. If one were concerned with the properties of mixtures, then the mixture identity might be the hub from which all other data stems, or one might choose to focus on individual measurements first and describe what they relate to as a subsidiary. Correct choice of the central identity makes our thinking about data access more simple and reduces the likelihood of designing an ontology that excludes future possibilities. The sum of our ontology for chemical entities and their properties is shown in Figures 1−4.

An example use of the ontology is provided in RDF/XML format in Appendix A, describing several useful properties of a crown ether.

It is vital to note that, in a free-form RDF data structure, none of the triples are compulsory nor are we limited to just one of each. A chemical entity can have any number of properties assigned to it, as well as many names, but it needs only one empirical formula. Also, although no element is compulsory, it is important that most items are included to allow differentiation between entries. It is inappropriate to store molecular properties in this system if the chemical entity had no unique identifier to locate it, or if there were no record of the place from which the property came.

At the root of this schema is a node that identifies each chemical entity uniquely. Originally, it was intended for the InChI[11] of each molecular structure to be both the central

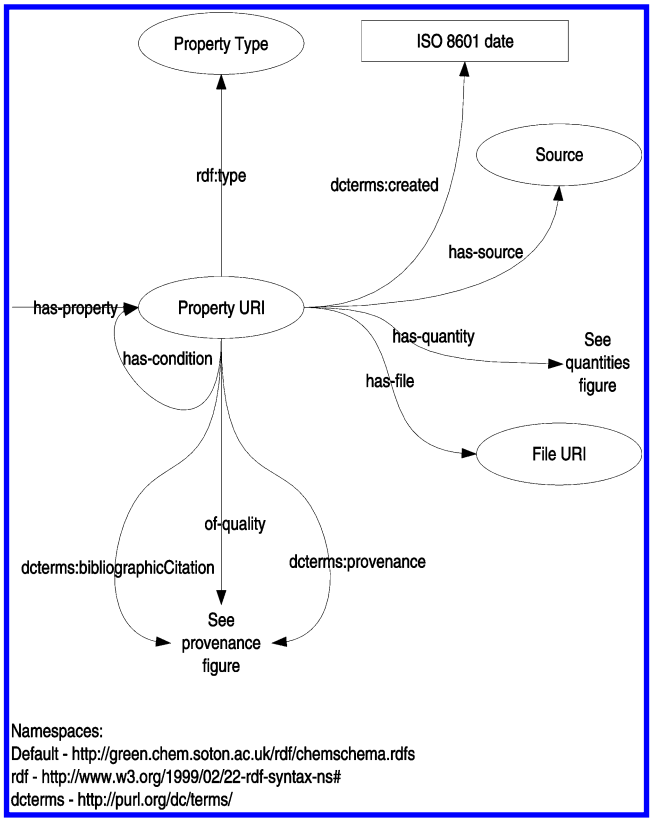

Figure 2. Schema layer 2: condition-dependent properties.

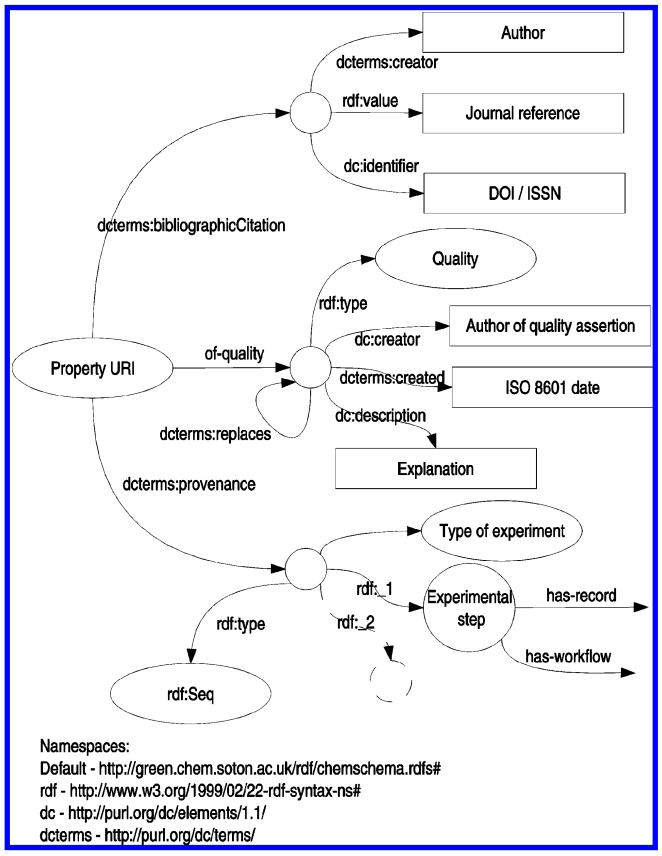

Figure 3. Schema layer 3: provenance of properties.

node and a searchable identifier. An InChI is a character string that describes a molecule on the basis of its structure. Unfortunately, there was a problem with this—InChI strings can become very long and contain characters that are often reserved for operating system use and, thus, present problems

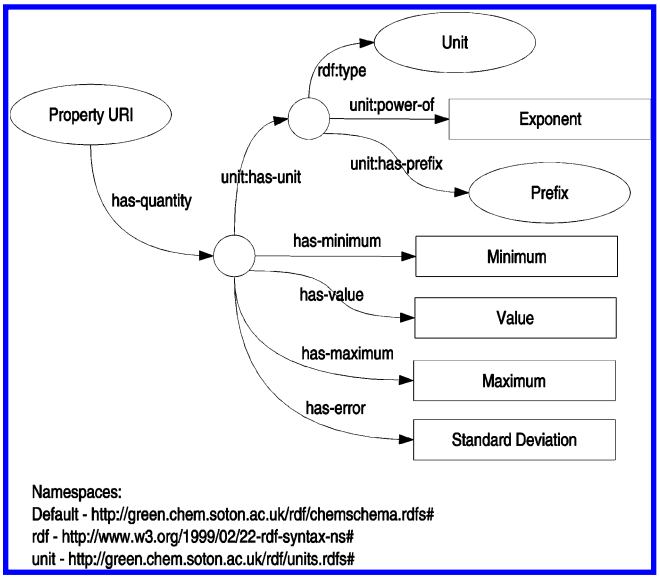

Figure 4. Schema layer 3: describing numerical values.

as a primary index. Our original intention to use the InChI as a filename for the RDF files rapidly created difficulties with referring to the files. The potential length of InChI strings exceeds practical limitations in some database software, and this also inhibited our development. To circumvent this, an SHA1 hash[23] was taken of the string. An SHA1 hash is a fixed length hexadecimal code that is computed from other data and is commonly used for integrity checking of data. Here, it serves well as a central node identifier since it is directly derived from the InChI, rather than just using some randomly assigned number. There is a possibility for two files to produce the same SHA1 code, but 2^{160} possible permutations suggest that this is sufficiently unlikely (a variation of the "birthday paradox"). This problem has not been encountered with 2.7 million InChI's generated from the ZINC database.[24] Should it become an issue, it is feasible to abandon the hashed InChI for a stand-alone URI or to reinstate the full InChI, but this would require some significant software changes.

Extending from this central node are two distinct layers of information. The first is depicted in Figure 1 and contains information about the chemical entity that is independent of state and conditions. This includes chemical names (systematic and trivial), reference codes from popular databases, and any classification for the chemical species, for example, organic, inorganic, or organometallic. Such classifications offer convenient filtering when attempting to construct data sets. Chemical names present a problem to computerized text handling, owing to a large number of characters from various languages; hence, the unicode format is required. More complicated nomenclatures such as with polymers require some form of substitution for super- and subscripts, since from a computerized perspective these details are formatting and, therefore, do not receive their own special symbols. Under these circumstances, a form of markup such as HTML is required to retain the original name, but this comes at the expense of requiring software capable of rendering the names and also makes name searching somewhat more complicated. This is unavoidable under the present character encoding methodology.

The second layer depicted in Figure 2 is more complex. The property URI acts as an anchor point for all information

about a property whether it be a physical property such as melting point or a spatial geometry recorded in a particular file format. Should one property be dependent on another (an experimental condition), or derived from some other property (such as a molecular surface area dependent on the geometry of the molecule), then links are made between the two property URIs. The property itself may be stored in a file or explicitly described in RDF depending on the uses for the data. In the case of a complex data collection such as a mass spectrum, it is far more sensible to make use of existing storage methods and software than to attempt to copy them with a less efficient RDF and have to create new tools to read them. Under these circumstances, the RDF becomes a file index laden with information about the origins and content of the file.

A third layer shown in Figures 3 and 4 consists of provenance, numerical values, and quality assertions. All values must have a unit associated with them, so a means to describe those units in more than just plain text is provided. Numerical values are rarely just a single exact number; hence, scope must be provided for a range of values and some estimate of error. Here, only a quotation of a standard deviation is given, but similar statistics can be attached with minimal effort. The provenance of a particular property comes in the form of a possible literature citation, an assertion of the quality of data, and a series of steps taken to create the data. Again, the representation of citations is basic and functional for reference purposes but could be extended to something as full as that found in ThermoML specifications.

Data can be classified by their quality, with specific reference to when values are found to be erroneous. These data can then be classified as bad while still retaining the record. Each new quality assertion supersedes its predecessor such that it is possible to have a chain of opinions and reasons without disposing of or compromising the original value. When such a quality issue is discovered, the "derived-from" links between property URIs can be followed to deduce what other properties may have been influenced by this bad property.

Last, a chain of steps that led to the property can be described using the RDF sequence construct. This may take the form of a series of programs and their inputs making up a computational workflow, or a series of experiments conducted one after the other. In the future, we would hope that, for experimental data, a link would be made to an electronic notebook entry, or similar digital record.

This combination of metadata and data should be sufficient to provide a trail of information by which individual data points can be traced back to their original source and reproduced if need be. Such point-by-point inspection has rarely been possible in existing databases, and even then, such verification has rarely gone beyond a simple journal reference.

This scheme for describing chemical data is the product of several iterations of development in which the starting ideas were gradually fleshed out in more detail until everything required could be described in a form suitable for software processing. A key benefit of using RDF for this purpose is that existing data can have new data added to it following the improved schema as long as the fundamental structure does not change. The need to impose some structure on the underlying RDF statements in order to handle the input into the triplestore suggested that, in the first instance, all information should be grouped by the chemical entity to which it applies; that is, the head node would be a unique identifier for the chemical entity. This is not as simple a choice as it seems as what chemists view as the same chemical species depends on the context. While the InChI provides a very useful method of generating a URI for chemical entities from their structure, it is in some ways too specific and yet not specific enough when deciding where to attach new properties.

To take a particular example of isomerism in molecular structure (same chemical formula and same atoms but arranged differently in connectivity and spatial arrangement), because of a possible pharmaceutical context for the work (e.g., drug design), the possibility of enantiomeric forms of molecules (the pair of molecules that are simply mirror images but, therefore, have very different biological properties) must be considered. It is not always possible to tell which enantiomer or mixture of enantiomers were present in semantically poor legacy data. Some chemical properties are common to both enantiomers and so need to be inherited by both entries in the database, while optical activity must be kept separate.

A similar problem exists with isotopic abundances. Isotopically labeled molecules are differentiated by the InChI even though the majority of their properties remain the same within experimental accuracy. It is largely a matter of judgment and individual need whether properties from the normal chemical entity should be linked to isotopic variants. Although it is possible to perform less specific InChI matching, in practice, this may present problems of knowing precisely which properties from which enantiomers and variants to present when relevant information is requested. In addition to this, the InChI is still a newly developed system and does not behave perfectly with respect to inorganic species. In other words, the InChI of a structure is not a perfect URI for this purpose, but with further improvement and careful handling, it should serve its purpose.

Considering materials as well as the constituent molecules means that polymorphism (that the same molecules can adopt different 3D packing structures when forming a crystal that can dramatically alter their macroscopic properties) needs to be considered. However, no molecular-based chemical identifier can currently capture polymorphic information. Nor indeed do most databases of measurements even provide information about the polymorph they refer to. It is assumed that only one polymorph exists until proven otherwise. We have adopted a system that directly associates the polymorph with the 3D crystal structure. Where properties relate to a particular polymorph, the properties themselves are linked to structures that describe the polymorphs rather than using some nomenclature or more elaborate hierarchy to relate them.

The next consideration was the handling of properties. Several properties useful for indexing (molecular weight and reference codes of other databases such as those from the Cambridge Crystallographic Data Centre, CCDC) are independent of where they came from, and so it is unnecessarily inefficient to build them into the same system that accommodates properties with the associated baggage of provenance needed for experimentally determined quantities. It

Chart 3

```
<ch:has-property>
 <rdf:Description rdf:nodeID="property01">
  <rdf:type rdf:resource="ch:Structure"/>
  <dcterms:created>2004-10-07T10:22:41Z</dcterms:created>
  <ch:has-source rdf:resource="sources:CCDC"/>
  <ch:has-file>
   <rdf:Description rdf:nodeID="file01">
    <ch:has-path>file://filestore/OHBOCP.cif</ch:has-path>
    <rdf:type rdf:resource="ch:CIF"/>
   </rdf:Description>
  </ch:has-file>
 </rdf:Description>
</ch:has-property>

<ch:has-property>
 <rdf:Description rdf:nodeID="property02">
  <rdf:type rdf:resource="ch:Structure"/>
  <dcterms:created>2005-07-20T10:23:15Z</dcterms:created>
  <ch:has-source rdf:resource="sources:NCI"/>
  <ch:has-file>
   <rdf:Description rdf:nodeID="file02">
    <ch:filename>file://filestore/molfiles/benzocrownether.mol</ch:filename>
    <dcterms:created>2001-02-07T13:15:25Z</dcterms:created>
    <rdf:type rdf:resource="ch:Mol"/>
   </rdf:Description>
  </ch:has-file>
 </rdf:Description>
</ch:has-property>
```

was decided that these properties that are dependent only on chemical structure should be separated out to make use of their value as filters for subset selection. Three-dimensional structures were not included in this, as they are dependent on the method used to obtain the structure, such as which force field was applied to a calculation, or if it was an X-ray or neutron method of structure determination. Associating 3D structures with methods allows several structures to be given for a chemical entity, such as those generated from X-ray diffraction data and those produced by high-level quantum simulations. Further calculations may be performed, in which case, it is vital we know which structure they began from, should we need an explanation for an unusual result. Some structure files have an internationally agreed naming convention that gives away their application—CIF files are molecular structures from an X-ray diffraction experiment. No further explicit definition of file content or form is needed within the store.

For a full RDF description of a crown ether with several properties, see Appendix A. RDF is not intended for human consumption, and the forms presented here have been written in their most legible form rather than the more terse but largely incomprehensible styles produced by triplestore software. An example of two simplified structure properties in pretty-printed RDF is shown in Chart 3. (Fully qualified namespaces are omitted for brevity.)

Yet another division that was considered was related to the phase of a property. For example, one may have a density of a substance in any of the common states of matter, or computed structures may be created in a vacuum or a solvent shell. These must be differentiated somehow, and it would be obvious to partition physical properties into phases that they apply to, but this distinction was not made here. Phase information is defined by supplementary data about properties. Properties in which phase is important should be stored alongside the conditions that control the phase, such as temperature or pressure, but this is not always the case in existing data sources. If the melting or boiling point of a compound is not already known, then there is no way of deciding which phase to put an existing record in, and hence, this facet would be left unknown. The vast majority of available data would lead to this kind of uncertainty, and so this issue has been left for further consideration in the future. If too much of a hierarchy is introduced, it becomes a more and more intricate process to extract data, which is naturally undesirable in database applications.

This series of decisions has produced a new approach applied to data capture and storage; it is possible to filter

Chart 4

```
<ch:Quantity>
 <ch:has-value>110</ch:has-value>
 <unit:has-unit>
  <unit:Unit>
   <rdf:type rdf:resource="unit:Joule"/>
   <unit:prefix rdf:resource="unit:Milli"/>
   <unit:power-of>1</unit:power-of>
  </unit:Unit>
 </unit:has-unit>
 <unit:has-unit>
  <unit:Unit>
   <rdf:type rdf:resource="unit:Kelvin"/>
   <unit:power-of>-1</unit:power-of>
  </unit:Unit>
 </unit:has-unit>
</ch:Quantity>
```

data on the basis of author, data source, method, accuracy, conditions, and molecular properties such as relative molecular mass. None of these filterable properties are particularly new, but together they far exceed the scope of presently available products. The descriptive power of RDF allows all of these details to have equal importance within software, and triplestore software allows those details to be used as search criteria. Even more usefully, abnormal data can be isolated and the supplementary information examined for possible reasons for its abnormality. If it should appear anomalous and the original data proved incorrect, it can be marked as untrustworthy. It is important to stress that even incorrect values need not be discarded in case it was the judgment of the value that was in error and not the value at all. An overtly incorrect experimental value might be recreated in the laboratory and the new correct value placed in the database, but the old value is not lost, merely superseded. A trail of precedence is maintained that guarantees we keep the original data but select the newer value by preference.

We have also found it necessary to develop a units system to support the chemical data by providing a manageable way to make scientific units machine-parseable. The expression of units for software use is a complicated matter and will be discussed in a separate publication, so only an overview is given here, along with an example of the storage of the units of a measurement. RDF is used to create a network of units and quantities that can be extended with new units and conversions without requiring any rewritten software. It has several advantages over the existing XML methods by rigorously limiting the ways in which units relate to each other, and by clearly addressing issues of dimensionality, convenience, and functionality. It follows a philosophy of minimalism, such that maintenance of the libraries is as simple as possible while providing all the necessary information to perform useful operations with units. To make these improvements, small sacrifices are made in the length of data description, and those who use it must become aware of the additional complexities of describing scientific data correctly.

Its invocation takes the form given in Chart 4. The power of this RDF expression is not immediately apparent, as it merely shows a method of recording the units of a value. The presence of an external ontology that maps the relationships between units provides the facility to automatically convert between units.

Capturing all this detail in a traditional database is a significant challenge. RDF provides a solution to many of these problems by supplying flexibility and variable structure at the cost of performance. An RDF graph can be extended at will without disrupting the whole system. In this sense, an RDF triplestore can act as a prototype database in which the problems of data description are discovered and dealt with prior to building a full production system. If speed is the ultimate concern, as opposed to semantically loss-free storage, then the problem is well-understood and conventional RDBMS solutions can be applied with the RDF converted into a relational model.

A major criticism of RDF and triplestores in general is their inefficiency —RDF syntax is very verbose, and its flexibility requires the description of relationships unnecessary in more common description methods. Using the schema presented here for describing chemical entities, it requires approximately 30 triples to capture all of the information regarding one entity and one property of that entity. Usefully sized data sets in chemoinformatics range from trivial hundreds of compounds to virtual screening libraries of several million compounds. The NCI open structures database[25] contains more than 250 000 unique species and is representative of a such a screening library, while the ZINC[24] database represents a much larger docking library consisting of more than 3 million structures. With numbers of this scale, such inefficiency might present a serious problem.

3.1. Scaling RDF Up. Memory-based triplestores are unsuited to handling large knowledge stores. They are transient and rapidly outgrow available computing resources. A computer with 1 GB of memory can handle no more than 1.1 million triples using the Redland[26] in-memory triplestore. With these constraints, it would be possible to describe only one-eighth of the NCI data using the proposed schema, and with no room for growth. The persistent triplestore is a relatively new type of software that stores RDF efficiently on nonvolatile storage media such that it can be used as a database in the same manner as traditional relational database systems such as Oracle and various flavors of SQL servers. These persistent triplestores are not constrained by memory limitations in the same way as volatile triplestores.

In Comb*e*Chem, we have experimented with three persistent RDF triplestores:

Jena is a Java framework for building Semantic Web applications, available from HP Labs[27] as open source software under a BSD license. Jena implements APIs for RDF and OWL and, using JDBC, can couple with existing RDBMSs (relational databases) such as MySQL or PostgreSQL or store triples in system memory. It offers RDFS (RDF schema) reasoning over in-memory stores and RDQL queries.[28]

3store is a set of tools built on a core C library that uses MySQL to store its raw RDF data and indices and is available under the GPL.[29] It also supports RDFS reasoning, can communicate using a variation of the "Open Knowledge Base Connectivity" (OKBC) protocol, and can answer RDQL queries.

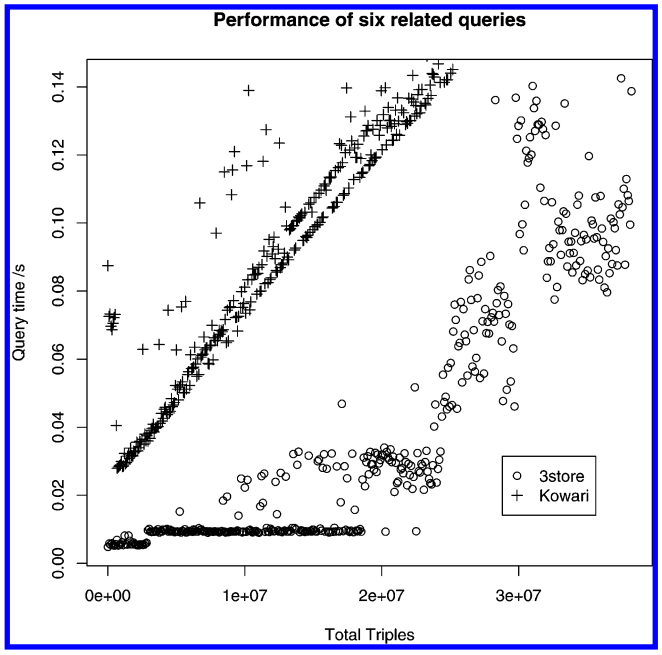

Figure 5. Scalability testing of triplestores.

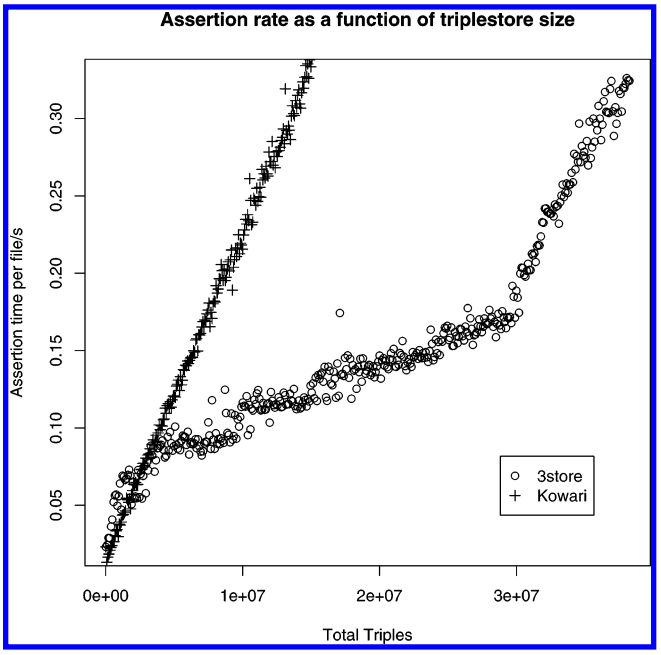

Figure 6. Scalability testing of triplestores.

Kowari is a Java-based triplestore available under the Mozilla Public License from Tucana Technologies.[30] It does not rely on an external RDBMS to provide the actual store and supports queries in its own query language called iTQL in addition to a raft of other access methods.

The authors could only find one objective assessment of triplestore capabilities,[31] but the author used a comparatively small data set (279 337 triples), and so no reliable evidence exists for truly large-scale triplestore use. To decide whether persistent triplestores could handle the sizes of data required for useful chemical storage, a number of tests were run on commodity hardware with two of the stores: 3store and Kowari. Jena was not included in this test because it is not a stand-alone database program but rather a library to add triplestore capabilities to the software that invokes it.

The test system was as follows: **CPU**: 2 × Opteron 246. **Memory**: 4 GB. **Disk**: 5 10K rpm SCSI disks in software RAID 5 array. **OS**: 64-bit Linux (Debian Stable). **Software**: Java 1.4.2, MySQL 4.0.24, Kowari 1.1.0-pre2, 3store 2.2.22.

Both the MySQL back end and Kowari were allocated sufficient memory to make good use of available hardware, and the server itself was devoted to the task of running each triplestore exclusively while testing was performed.

A test data set was constructed consisting of 36 million RDF triples distributed into files, each consisting of a description of a proportion of unique molecules from the ZINC database. The triplestores were tested for response times to a chain of six queries emulating a walk around an RDF graph, and the time taken to insert each additional file of RDF was measured. Figures 5 and 6 show how the quantity of data affected the speed of the systems.

Kowari behaves in a consistent manner, with query time scaling linearly within the range of the data. 3store displays several distinct regimes relating to the gradual saturation of caching mechanisms within the MySQL server. With small numbers (less than 2 million) of triples, the response time is almost instantaneous because the data resides entirely within memory. The second regime stretching up to 20 million triples shows no significant increase in response time except in few cases where much slower responses are observed. From 20 million triples upward, all caching advantages are exhausted, and 3store adopts a similar response scaling to Kowari albeit in a more noisy fashion. Both triplestores respond to queries with surprising variation, probably as a result of the state the system is in when a query is received, or other operating system processes being performed such as file system journaling. Sometimes, the necessary data will be immediately available, while other times, it must be retrieved from disk with the corresponding performance penalty. It should be noted that more conventional desktop machines will reach these less favorable scaling regimes much sooner due entirely to memory limitations. Further optimizations of server and software settings may further improve performance.

The largest overhead to operating a triplestore is the assertion of new triples into the knowledge base, although our approach of many small files is distinctly suboptimal. Figure 6 shows the same linear behavior from Kowari and three distinct regimes from 3store. When dealing with small data sets (less than 1 million triples), Kowari asserts more quickly, but 3store and MySQL settle into a more favorable behavior that is maintained up to 30 million triples. Again, a sharp dog-leg appears, matching the gradient of the Kowari line, but by this stage, many hours have been saved, giving 3store a clear advantage. Although the response times for both query and assertion using 3store are less consistent and predictable, the absolute magnitudes are favorable when compared with Kowari.

3store was adopted for this work because it has been shown to be capable of handling volumes of triples needed for this application and scales more favorably than the other triplestore tested here. Additionally, it is easily batch- or Perl-scriptable, supports RDFS scalably, and can use standard database tools for the maintenance of data (e.g., backups and migration) as all application state data is held in the database back end. Kowari, by contrast, must use its own less well-proven tools to perform these operations.

Chart 5

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
 xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:ch="http://green.chem.soton.ac.uk/rdf/chemschema.rdfs#"
 xmlns:dc="http://purl.org/dc/elements/1.1/"
 xmlns:dcterms="http://purl.org/dc/terms/"
 xmlns:sources="http://green.chem.soton.ac.uk/rdf/sources.rdfs#"
 xmlns:unit="http://green.chem.soton.ac.uk/rdf/units.rdfs#"
 xmlns:prov="http://green.chem.soton.ac.uk/rdf/provenance.rdfs#">
<ch:Molecule rdf:about="file://filestore/9ef95c6b959d0211d339ea942e21ba7a.rdf">
 <ch:has-inchi>C14H20O5/c1-2-4-14...12-19-14/h1-4H,5-12H2/b9-5-,10-6-</ch:has-inchi>
 <ch:has-simple-inchi>C14H20O5/c1-2-4-14...12-19-14/h1-4H,5-12H2</ch:has-simple-inchi>
 <ch:has-rmm>268.3056</ch:has-rmm>
 <ch:has-name>BENZO-15-CROWN-5-ETHER</ch:has-name>
 <ch:has-name>2,3,5,6,8,9,11,12-Octahydro-1,4,7,10,13-benzopentaoxacyclopentadecin</ch:has-name>
 <ch:has-cas>14098-44-3</ch:has-cas>
 <ch:has-empirical-formula>C14H20O5</ch:has-empirical-formula>
 <rdf:type rdf:resource="ch:OrganicMolecule"/>
 <ch:has-property>
  <rdf:Description rdf:nodeID="property01">
   <rdf:type rdf:resource="ch:Structure"/>
   <dcterms:created>2004-10-07T10:22:41Z</dcterms:created>
   <ch:has-source rdf:resource="sources:CCDC"/>
   <ch:has-file>
    <rdf:Description rdf:nodeID="file01">
     <ch:has-path>file://filestore/OHBOCP.cif</ch:has-path>
     <rdf:type rdf:resource="ch:CIF"/>
    </rdf:Description>
   </ch:has-file>
   <dcterms:biblographicCitation>
    <rdf:Description rdf:nodeID="a31c3">
    <dc:creator>Simon J Coles</dc:creator>
    <dc:creator>Michael B Hursthouse</dc:creator>
    <dc:creator>Jeremy G Frey</dc:creator>
    <dc:creator>Esther Rousay</dc:creator>
    <rdf:value>http://ebank.eprints.org/145/</rdf:value>
   </rdf:Description>
  </dcterms:biblographicCitation>
  <dcterms:provenance>
   <rdf:Description rdf:nodeID="a31c2>
    <rdf:type rdf:resource="ch:Laboratory"/>
    <!-- Link to Electronic Lab Notebook via URI -->
   </rdf:Description>
  </dcterms:provenance>
  <ch:of-quality>
   <ch:Quality rdf:nodeID="a31b8">
    <rdf:type rdf:resource="ch:Good"/>
    <dc:creator>Kieron Taylor</dc:creator>
```

Chart 5 (continued)

```
    <dcterms:created>2005-07-20T17:57:39Z</dcterms:created>
   </ch:Quality>
  </ch:of-quality>
 </rdf:Description>
</ch:has-property>
<ch:has-property>
 <rdf:Description rdf:nodeID="property02">
  <rdf:type rdf:resource="ch:Structure"/>
  <dcterms:created>2005-07-20T10:23:15Z</dcterms:created>
  <ch:has-source rdf:resource="sources:NCI"/>
  <ch:of-quality>
   <ch:Quality rdf:nodeID="a31b9">
    <rdf:type rdf:resource="ch:Good"/>
    <dc:creator>Kieron Taylor</dc:creator>
    <dcterms:created>2005-07-20T17:57:39Z</dcterms:created>
   </ch:Quality>
  </ch:of-quality>
  <ch:has-file>
   <rdf:Description rdf:nodeID="file02">
    <ch:filename>file://home/molfiles/9/e/f/9ef95c6b959d0211d339ea942e21ba7a.mol</ch:filename>
    <rdf:type rdf:resource="ch:Mol"/>
   </rdf:Description>
  </ch:has-file>
  <dcterms:provenance>
   <prov:Provenance>
    <rdf:type rdf:resource="ch:Calculated"/>
    <rdf:type rdf:resource="rdf:Seq"/>
    <rdf:_1>
     <prov:Step>
      <ch:has-description>http://green.chem.soton.ac.uk/methods/ncicorina.htm</ch:has-description>
     </prov:Step>
    </rdf:_1>
   </prov:Provenance>
  </dcterms:provenance>
 </rdf:Description>
</ch:has-property>
<ch:has-property>
 <rdf:Description rdf:nodeID="property03">
  <rdf:type rdf:resource="ch:PartitionCoefficient"/>
<dcterms:created>2005-07-21T19:59:45Z</dcterms:created>
<ch:has-source rdf:resource="sources:PhysProp"/>
<ch:has-quantity>
 <ch:Quantity rdf:nodeID="a31bb">
  <ch:has-value>0.91</ch:has-value>
  <unit:has-unit>
   <unit:Unit rdf:nodeID="a31bc">
    <unit:power-of>1</unit:power-of>
    <rdf:type rdf:resource="unit:Mol"/>
   </unit:Unit>
```

Chart 5 (continued)

```
  </unit:has-unit>
  <unit:has-unit>
   <unit:Unit rdf:nodeID="a31bd">
    <unit:power-of>-1</unit:power-of>
    <rdf:type rdf:resource="unit:Mol"/>
   </unit:Unit>
  </unit:has-unit>
 </ch:Quantity>
</ch:has-quantity>
 <ch:of-quality>
  <ch:Quality rdf:nodeID="a31be">
   <rdf:type rdf:resource="ch:Good"/>
   <dcterms:created>2005-07-21T19:59:45Z</dcterms:created>
   <dc:creator>Kieron Taylor</dc:creator>
  </ch:Quality>
 </ch:of-quality>
 <dcterms:bibliographicCitation>
  <prov:Citation>
   <rdf:value>STOLWIJK,TB ET AL. (1989) - Consult Physprop supplementary information</rdf:value>
  </prov:Citation>
 </dcterms:bibliographicCitation>
 <dcterms:provenance>
  <prov:Provenance rdf:nodeID="a31bf">
   <rdf:type rdf:resource="ch:Laboratory"/>
  </prov:Provenance>
 </dcterms:provenance>
 <ch:has-condition>
  <rdf:Description rdf:nodeID="property04">
   <rdf:type rdf:resource="ch:Temperature"/>
   <ch:has-quantity>
    <ch:Quantity rdf:nodeID="a31c0">
     <ch:has-value>25</ch:has-value>
     <unit:has-unit>
      <unit:Unit rdf:nodeID="a31c1">
         <rdf:type rdf:resource="unit:Celsius"/>
         <unit:power-of>1</unit:power-of>
        </unit:Unit>
       </unit:has-unit>
      </ch:Quantity>
     </ch:has-quantity>
    </rdf:Description>
   </ch:has-condition>
  </rdf:Description>
 </ch:has-property>
</ch:Molecule>
</rdf:RDF>
```

4. RESULTS AND CONCLUSIONS

A simple Web interface has been constructed that allows the underlying RDF technology to behave as a conventional database. All data about a particular chemical species can be returned in its structured form from a collection of hundreds of thousands of species. Queries can be made using any of the chief molecular identifiers (InChI, CAS number, name, etc.), and the page of information provided includes

renderings of any 3D structures linked to by the triplestore. This illustrates the aggregation of information from multiple data sources into one dynamically generated reference page. Even with an 80 million triple knowledge base, this exploration of the RDF remained rapid enough for reference use.

The number of different indices available makes this a useful resource for general chemical reference. It is also one of the first databases to make use of the InChI in the presence of more common chemical identifiers and, thus, acts as a bridge from one identifier system to another. Last, the RDF can be exported to other triplestores (both in-memory and persistent) with minimal effort and put to whatever use may be appropriate.

Underneath the simple interface is the powerful query capacity of the triplestore, whereby subsets of scientific data can be selected that were recorded on particular dates using particular methods, differentiating between experimental and predicted results such that it is known where each data point came from as well as how reliable it is. Where one result has been derived from another, the extensive provenance can tell us which results created the present entry and, thus, deduce the knock-on effects of a correction on the old data. With appropriate workflow enactment, it will be straightforward to rerun whatever processes were used to produce the present entry and obtain a new answer.

Predictive model building in, for example, QSAR or QSPR studies pivotal to virtual screening have now reached a level of complexity such that a wide range of chemical descriptors is needed to attempt to describe even a small proportion of chemical space. New descriptors are being invented at a rapid pace, pushed by the increasing ability to calculate them from basic structural data, using of course the increasing availability of computer power. These descriptors need to be made available for subsequent model building, and the community will benefit from the ability to make these descriptors available once calculated, thereby saving computation time. Similarly, once the model building is underway, there is a great need to be able to link back to the raw data available about the set of chemical entities used to build the model, to understand the ever-present outliers, some of which will in fact be due to poor original data, thus the need for provenance data as well as the descriptor values.

The nature of the technologies used to implement this system make it comparatively easy to link together multiple distinct triplestores. The use of an overriding ontology, the presence of URIs in the data, and the compartmentalization of data into distinct RDF files about individual chemical entities provide the basis for multiple databases to answer the same queries and combine their answers into a greater amalgamation. This allows control of access to proprietary data without isolating it from the greater data grid and also removes the need for a centralized collating system to give universal access to data. For example, a crystallography department may provide a structures triplestore, while a computational department serves up the results of their simulations. A client may then query both systems and receive knowledge of data that may have otherwise escaped notice. This sort of data discovery reflects the Semantic Web vision applied to chemical research, as increasing participation increases the power of the method.

Although still a new technology, triplestores have now reached a state of usability. Further addition of chemical properties to our test system is an ongoing process, which increases its value for data mining. The future development of automated calculations using the many available structures and the subsequent storage of the results alongside all the details of the computations that produced them will be a significant aid to the QSPR development process. Beyond that, it should be possible to achieve high-throughput data processing and further accelerate model development.

3store has been subjected to a very large bulk of data conforming to the schema presented here and remains responsive, but data import performance continues to degrade with increasing size. Write performance on large stores is known to be a challenging issue, and it is likely that a single large triplestore with frequent insertions would be unable to cope with potential demand. The quantity of data in our tests could easily be doubled or trebled when populating with computed properties, owing to the potency of high-throughput computation. Serious consideration of the matter of greater scale needs to be taken, including further optimization of the triplestores themselves and distributing and maintaining data across multiple stores.

The Semantic Web is an ambitious goal, and having chemical information there even more so, requiring contributions from multiple players in order to achieve maximum benefit. In chemistry, only a comparatively small population is interested in any particular area. One can conceive of free data exchange and banishment of the proprietary file format, but there are parties who do not want to make data more easily available to their competitors. However, this work demonstrates the value of adopting this approach on the scale of this project. The molecular properties triplestore plays an important part in building the Comb*e*Chem Semantic Datagrid by providing enhanced recording, storage, and retrieval of scientific data.

ACKNOWLEDGMENT

The authors would like to thank the EPSRC for funding this work.

APPENDIX A. A SAMPLE RDF FILE

A complete RDF file to describe a crown ether with several properties and structure files associated with the chemical entity is shown in Chart 5. The file, also supplied as Supporting Information, may be read into an RDF visualization tool such as the W3C validator found at http://www.w3.org/RDF/Validator/. This provides a view that reflects the graph of information encoded.

Supporting Information Available: The content of Appendix A can be downloaded (in .txt format) as Supporting Information for perusal using RDF-aware software. This material is available free of charge via the Internet at http://pubs.acs.org.

REFERENCES AND NOTES

(1) Frey, J. G.; Bradley, M.; Essex, J.; Hursthouse, M. B.; Lewis, S. M.; Luck, M. M.; Moreau, L.; De Roure, D.; Surridge, M.; Welsh, A. In *Grid Computing: Making the Global Infrastructure a Reality*; Berman, F., Ed.; Wiley: New York, 2003; Chapter 42: Combinatorial Chemistry and the Grid.

(2) World Wide Web Consortium, Resource Description Framework. http://www.w3.org/rdf/ (accessed 2005).

(3) Hey, T.; Trefethen, A. E. The UK e-Science Core Programme and the Grid. *Future Gener. Comput. Syst.* **2002**, *18*, 1017−1031.
(4) Frey, J. G.; De Roure, D.; Carr, L. A. Publication at Source: Scientific Communication from a Publication Web to a Data Grid. In *Euroweb 2002 Conference, The Web and the Grid: From e-Science to e-Business*, Oxford, U. K., December 17−18, 2002; Hopgood, F. R. A., Matthews, B., Wilson, M. D., Eds.; British Computer Society: Swindon, U. K.; Electronic Workshops in Computing.
(5) Berners-Lee, T.; Hendler, J.; Lassila, O. The Semantic Web. *Sci. Am.* **2001**, *284*, 34−43.
(6) De Roure, D.; Jennings, N.; Shadbolt, N. *Research Agenda for the Semantic Grid: A Future e-Science Infrastructure*; Technical Report UKeS-2002−02; National e-Science Centre: Edinburgh, U. K., 2001.
(7) De Roure, D.; Jennings, N. R.; Shadbolt, N. R. The Semantic Grid: Past, Present, and Future. *Proc. IEEE* **2005**, *93* (3), 669−681.
(8) World Wide Web Consortium (http://www.w3.org/) W3C Semantic Web Activity, Semantic Web Activity Statement. http://www.w3.org/2001/sw/Activity (accessed 2005).
(9) Goble, C. A.; De Roure, D.; Shadbolt, N. R.; Fernandes, A. F. A. A. Enhancing Services and Applications with Knowledge and Semantics. In *The Grid 2: Blueprint for a New Computing Infrastructure*; Foster, I., Kesselman, C., Eds.; Morgan-Kaufmann: San Francisco, CA, 2004; pp 431−458.
(10) Liefeld, T.; Martin, S.; Clark, T. Globally distributed object identification for biological knowledgebases. *Briefings Bioinf.* **2004**, *5*, 59−70.
(11) InChI International Chemical Identifier, IUPAC, http://www.iupac.org/inchi/. See also the unofficial InChI FAQ, http://wwmm.ch.cam.ac.uk/inchifaq/.
(12) SMILES: Simple and comprehensive chemical nomenclature. Daylight Chemical Information Systems, Inc: Aliso Viejo, CA. http://www.daylight.com/dayhtml/smiles/.
(13) World Wide Web Consortium (W3C), Extensible Markup Language. http://www.w3.org/XML/ (accessed 2005).
(14) Open Geospatial Consortium, GML − the Geography Markup Language. http://www.opengis.net/gml/ (accessed 2005).
(15) Cuellar, A. A.; Lloyd, C. M.; Nielsen, P. F.; Bullivant, D. P.; Nickerson, D. P.; Hunter, P. J. An Overview of CellML 1.1, a Biological Model Description Language. *Simul. Trans. Soc. Model. Simul. Int.* **2003**, *79*, 740−747.
(16) Murray-Rust, P.; Rzepa, H. S. Chemical Markup, XML, and the Worldwide Web. 1. Basic Principles. *J. Chem. Inf. Comput. Sci* **1999**, *39*, 928−942.
(17) Frenkel, M.; Chirico, R. D.; Diky, V. V.; Dong, Q.; Frenkel, S.; Franchois, P. R.; Embry, D. L.; Teague, T. L.; Marsh, K. N.; Wilhoit, R. C. ThermoML − An XML-based approach for the storage and exchange of experimental and critically evaluated thermophysical and thermochemical property data. 1. Experimental Data. *J. Chem. Eng. Data* **2003**, *48*, 2−13.
(18) Schäfer, B. A.; Poetz, D.; Kramer, G. W. Documenting laboratory workflows using the Analytical Information Markup Language. *J. Assoc. Lab. Automation* **2004**, *9*, 375−381.
(19) Rühl, M. A.; Kramer, G. W.; Schäfer, R. SpectroML − A Markup Language for Molecular Spectrometry Data. *J. Assoc. Lab. Automation* **2001**, *6* (6), 76−82. Rühl, M. A.; Kramer, G. W.; Schäfer, R. *SpectroML − An Extensible Markup Language for the Interchange of Molecular Spectrometry Data*; NIST Interagency Report 6821; NIST: Gaithersburg, MD, 2001; p 74.
(20) W3C Semantic Web Activity, Web Ontology Language. http://www.w3.org/2004/OWL/ (accessed 2005).
(21) RSS Developers Working Group, RDF Site Summary. http://web.resource.org/rss/1.0/ (accessed 2005).
(22) Murray-Rust, P.; Rzepa, H. S.; Williamson, M. J.; Willighagen, E. L. Chemical Markup, XML, and the World Wide Web. 5. Applications of Chemical Metadata in RSS Aggregators. *J. Chem. Inf. Comput. Sci.* **2004**, *44*, 462−469.
(23) Secure Hash Standard, National Institue of Science and Technology. http://www.itl.nist.gov/fipspubs/fip180-1.htm (accessed 2005).
(24) Irwin, J. J.; Stoichet, B. K. ZINC − A Free Database of Commercially Available Compounds for Virtual Screening. *J. Chem. Inf. Comput. Sci.* **2005**, *45*, 177−182.
(25) National Cancer Institute, Frederick and Bethesda Data and Online Services; National Institutes for Health: Bethesda, Maryland. http://cactus.nci.nih.gov/, 2004.
(26) Beckett, D. Redland RDF Application Framework. http://librdf.org/ (accessed 2005).
(27) Jena − A Semantic Web Framework for Java. http://jena.sourceforge.net/ (accessed 2005).
(28) Seaborne, A. RDQL − A Query Language for RDF, W3C Member Submission. http://www.w3.org/Submission/RDQL/ (accessed 2005).
(29) Harris, S.; Gibbins, N. 3store: Efficient Bulk RDF Storage. In *Proceedings of the First International Workshop on Practical and Scalable Semantic Web Systems (PSSS2003)*, Sanibel Island, Florida, 2003; pp 1−15.
(30) Tucana Technologies Inc, Kowari Metadata Store. http://www.kowari-.org/ (accessed 2005).
(31) Lee, R. Scalability Report on Triple Store Applications. http://simile.mit.edu/reports/stores/ (accessed 2005).

CI050378M