A Semantic Datagrid for Combinatorial Chemistry

Kieron Taylor, Rob Gledhill, Jonathan W. Essex, Jeremy G. Frey,
Stephen W. Harris and David De Roure

***Abstract*—The CombeChem project has designed and deployed an e-Science infrastructure using a combination of Grid and Semantic Web technologies. In this paper we describe the datagrid element of the project, which provides a platform for sophisticated scientific queries and a rich record of experimental data and its provenance. This datagrid constitutes a significant deployment of Semantic Web technologies and we propose it as an example of a 'Semantic Datagrid'.**

***Index Terms*—Datagrid, Semantic Web, Semantic Grid, Combinatorial Chemistry, chemical structure**

I. Introduction

THE Power of the World Wide Web stems not only from the content of the Web but from the way in which it is interlinked, which leads to ease of use and provides the additional machine-processable knowledge that underlies powerful Web search engines. In this paper we describe a Datagrid that has similarly been created with very rich interlinking of scientific data, enabling powerful queries to be performed and providing a comprehensive record of the provenance of experimental data. The datagrid, which contains various forms of scientific information over multiple datastores, draws on the tools and techniques of the *Semantic Web* – we refer to it as a *Semantic Datagrid*.

The next section outlines the CombeChem project. We then introduce the Semantic Web and discuss the integration of Web and Grid technologies, an area of work known as the *Semantic Grid*. From these origins we discuss the design, development and deployment of the CombeChem Semantic Datagrid. In closing we discuss the outcomes of our work.

II. The CombeChem Project

Combinatorial chemistry is an example of a domain in which new experimental techniques massively accelerate, or indeed parallelise, the experimental process. The synthesis of new chemical compounds by combinatorial methods provides major opportunities for the generation of large volumes of new chemical knowledge. This is the principal drive behind the CombeChem e-Science project [1], which aims to enhance the correlation and prediction of chemical structures and properties by increasing the amount of knowledge about materials via synthesis and analysis of large compound libraries. Given the throughput of knowledge created through combinatorial chemistry, it is not plausible for every new compound to be the subject of a traditional scholarly publication by a scientist, since this would introduce a massive bottleneck – perhaps 80% of data would be left unprocessed. Thus CombeChem sets out to process a greater volume of data, and in the process it suggests a significant culture shift in the scientific process within this discipline. Handling the data deluge is the common characteristic of many of the projects in the e-Science programme [2].

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 of the investigation – this is described as 'publication at source' [3]. The creation of original data is accompanied by information about the experimental conditions in which it is created. There then follows a chain of processing such as aggregation of experimental data, selection of a particular data subset, statistical analysis, or modelling and simulation. The handling of this information 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 this data is made available for subsequent reuse in support of the scientific process, subject to appropriate access control.

While some of the 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 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. Both these later computational techniques require large computing resources: the former is more suitable for the traditional supercomputing environment but the latter scales well over

Manuscript received June 3, 2005. This work was supported in part by the UK Engineering and Physical Sciences Research Council under Grants 'Structure-Property Mapping: Combinatorial Chemistry and the Grid (CombiChem)' (GR/R67729) and 'Advanced Knowledge Technologies' (GR/N15764) .

K. Taylor, J. Essex, J.G. Frey and R. Gledhill are with the School of Chemistry, University of Southampton, Southampton SO17 1BJ, UK (email: krt1@soton.ac.uk, J.W.Essex@soton.ac.uk, J.G.Frey@soton.ac.uk, R.J.Gledhill@soton.ac.uk). J.G. Frey is the corresponding author (phone +44 23 8059 3209).

D. De Roure and S.W. Harris are with the School of Electronics and Computer Science, also at University of Southampton (e-mail: dder@ecs.soton.ac.uk, swh@ecs.soton.ac.uk).

Beowulf clusters. In both cases access to a grid infrastructure directly benefits the automation of the calculation workflows.

III. The Semantic Grid

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 [4] to Grid infrastructure and applications in the 2001 report 'Research agenda for the Semantic Grid: A future e-science infrastructure' [5]. The report drew on the CombeChem 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 many organizations, 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)

While the Grid provides the necessary distributed systems infrastructure, the Semantic Web provides the complementary capability with respect to distributed information. Hence the Semantic Grid enables scientists to answer questions, which involve integration of scientific data and automatic execution of computations, providing important functionality at the datagrid and scientific applications level. It also facilitates automation within the grid middleware – 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 [6].

Some of the Semantic Web's 'added value' comes from accumulating descriptive information (metadata) about the various artefacts 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, which is the standardised naming schema for biological entities in the Life Sciences domains, and the InChI (International Chemical Identifier [7]) which in chemistry provides a unique identifier for the molecular structure of each (organic) compound.

The W3C's Resource Description Framework (RDF) [8] enables the metadata, and hence the relationships between things, to be expressed in a machine-processable way. An RDF structure consists of a set of relationships, called triples. Each triple typically consists of three URIs: the subject and the object, which refer to two entities, and the predicate, which is a URI with a commonly agreed meaning, representing the relationship between the subject and the object. For example, to express 'the semanticgrid.org home page has a creator whose value is David De Roure' we could use an RDF triple consisting of

Subject	http://www.semanticgrid.org/index.html
Predicate	http://purl.org/dc/elements/1.1/creator
Object	http://www.ecs.soton.ac.uk/people/dder

The Semantic Grid depends upon making knowledge explicit and processable by machine, 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 conceptualisation 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. Furthermore, since ontologies encode relationships between classes of object, inferences can be drawn between instances; for example, reasoning can be achieved in the OWL standard using a variety of description logic inference engines. The website www.semanticgrid.org contains further information about Semantic Grid and the Global Grid Forum (GGF) Semantic Grid Research Group.

IV. Semantic Grid in CombeChem

The CombeChem project has been tackled from a Semantic Grid perspective. Automation of measurement and analysis is required in order to achieve the CombeChem requirements efficiently and reliably, and is a clear case for making knowledge explicit and machine processable through the application of Semantic Web technologies. Another role of Semantic Web technologies in this project – and our focus here – is to establish this complete chain of interlinked digital information all the way from the experiment through to publication.

This starts in the smart laboratory and Grid-enabled instrumentation [9]. By studying chemists within the laboratory, technology has been introduced to facilitate the information capture at this earliest stage [10]. Additionally, pervasive computing devices are used to capture live metadata as it is created at the laboratory bench, relieving the chemist of

the burden of metadata creation. This data then feeds into the scientific data processing. All usage of the data through the chain of processing is effectively an annotation upon it. By making sure everything is linked up through shared URIs, or assertion of equivalence and other relationships between URIs, scientists wishing to use these experimental results in the future can chase back to the source (i.e. the provenance is explicit).

Some of these ideas are also demonstrated in the World Wide Molecular Matrix [11] and the Collaboratory for Multi-scale Chemical Science (CMCS) [12].

V. Building the Semantic Datagrid

A. The role of RDF

XML is a generalised 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, biology and chemistry, and multiple programs can interpret these XML files. RDF represents an additional level of metadata above this, because it not only formalises what things 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, and more importantly it encourages the sharing of schemas (such as the ubiquitous Dublin Core), so that many tools can understand that an object in one document is the same as an object in another document.

The traditional relational database model demands a sizeable development period prior to the operation of the database. Exact requirements must be deduced, and a detailed database schema drawn up before the system can be implemented. Any mistakes or oversights made during this design process causes significant problems later on. If the schema cannot capture the desired data, the database must be rebuilt from the bottom, with the old data accordingly modified to fit within the new requirements. Adding an additional column of information about particular records can be a difficult task, and is in no way as simple as a typical spreadsheet representation may imply.

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 crystallised 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 don't care or don't know, but that does not excuse the data store from needing to specify these things. Too many data sources gloss over these details, relying on people not requiring 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 in line with the correct database design results in an exceedingly convoluted and unintuitive database 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 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.

Are we to take the easy solution, and limit the data we record to the existing storage format? RDF has the solution to many of these problems by providing flexibility and variable structure.

A pattern of triples can reflect any number of dimensions for any number of data types with ease. A second instance of a property for a particular molecule merely requires another set of triples rather than another layer of abstraction in the database. Where the relational database demands that all data fits the schema, an RDF schema can be redefined to accommodate new data beyond the original scope. The explicit nature of the triple – every data point requiring a declaration of what it is and what it relates to – may appear hugely inefficient by defining things in such a verbose manner, but this is what creates the flexibility. Nothing is assumed or left to the software to guess, so the limitation on data structures comes in the form of the software that deals with the output, and common sense for what should and should not be handled. If everything is logically spelled out in a net of interlinking data, the data is self-describing and can live on even without the database in which it is contained. This brings about the added implication that we can exchange our database software for a newer product while keeping the underlying knowledge.

In exchange for flexibility we lose speed. Relational databases have been developed and optimised for commercial use over three decades and are presently the fastest way of storing and retrieving large volumes of data. In contrast, RDF triplestores are a relatively new tool with neither a significant mass of software nor the abundant experience that make the deployment of such a technology easier. A triplestore must be walked by an algorithm due to the indeterminate nature of the data structure, and this is inherently slower than demanding particular data directly in a particular form.

The performance limitations of triplestores are not yet

known although they will never match relational databases with present technology and so we must compromise one way or the other: speed and simplicity versus flexibility. The choice is dependent on the function of the database. For chemical data, it is possible to supply so much supplementary information of future importance that flexibility appears to be the winner, technological considerations notwithstanding. There are more capable Semantic Web languages that RDF and RDFS, such as OWL, but these places higher computational burdens on the stores used to hold the data and execute queries. As efficiency is an issue, and RDFS and RDF appear to be sufficient to model this domain they were used in this instance.

B. RDF applied to chemical data storage

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 the emphasis could be placed elsewhere. 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 data access simpler and aids understanding of the data. Fig. 1 depicts the current schema, and an example fragment of RDF is provided in Appendix A.

It is vital to note that in a freeform RDF data structure, none of the objects are compulsory nor are we limited to just one of each. Multiple entries for some data items are meaningless, but a molecule can have any number of properties assigned to it. Also, although no element is compulsory it is important that most items are included to allow different items to be located easily. It would be foolish to store molecular properties if the molecule 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 molecule uniquely. Originally it was going to be the InChI (International Chemical Identifier) for that molecule. An InChI is a character string that describes a molecule based on its structure. Unfortunately, there was a problem with this – InChI strings can become very long and impractical as a primary index. To rectify this, an MD5 hash was taken of the string. An MD5 hash is a fixed length hexadecimal code that is computed from other data, and serves well as a central node identifier, rather than just using some randomly assigned number. There is a possibility for two files to produce the same MD5 code, but 2^{128} possible permutations suggest this is sufficiently unlikely (a variation of the 'birthday paradox').

Extending from this central node are two distinct layers of information. The first contains information about the molecule that is independent of state and conditions, while the second consists of all the information about properties where these factors are important. The second layer (marked Physical Property) makes up the bulk of the data and contains the provision for complete tracking of the origins of any data, as well as the data itself. Every value is not only associated with its molecule, but also an expression of uncertainty in that value, the units of the value, how trustworthy that value is (in case we should later discover a problem), the source of the data and the method by which it was obtained. This 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 to the computer. The need to impose some structure on the underlying RDF statements in order to handle the input into the triple store suggested that, in the first instance, all information should be grouped by the molecule to which it applies; i.e. the head node would be a unique identifier for the molecule. This is not as simple a choice as it seems as what Chemists view as the same molecule depends on the context. While the InChI provides a very useful method of generating a URI for a molecules from its structure, it is in some ways too specific to allow sensible chemically aware searching and indexing of molecules and their properties. A higher-level grouping was felt to be necessary to be able to generate a useful database.

To take a particular example of isomerism in molecular structure (same chemical formula, same atoms but arranged differently in connectivity of spatially), because of the pharmaceutical context for some of the work (e.g. drug design), we needed to consider the possibility of the enantiomeric forms of molecules (the pair of molecules that are simply mirror images but therefore have very different biological properties). The end user searching for a particular chemical structure will probably not specify the complete stereochemistry of the target molecule if they are working from a drawn structure or from a chemical name, so ambiguities of which enantiomer or other type of isomer could easily be present in the query.

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 which polymorph you have. Nor indeed do many databases of measurements even provide information about the polymorph they refer to. We have adopted a system that directly associates the polymorph with the 3D crystal structure.

The next consideration was handling of properties. Several properties useful for indexing (molecular weight, reference codes of other databases such as the Cambridge Crystallographic Data Centre, CCDC) are completely independent of where they came from, and so it is unwieldy to build them into the same system that accommodates properties with associated baggage of provenance that is needed for experimentally determined quantities. It was decided that these properties that are dependent only on chemical structure should be separated out to allow speedy location of data.

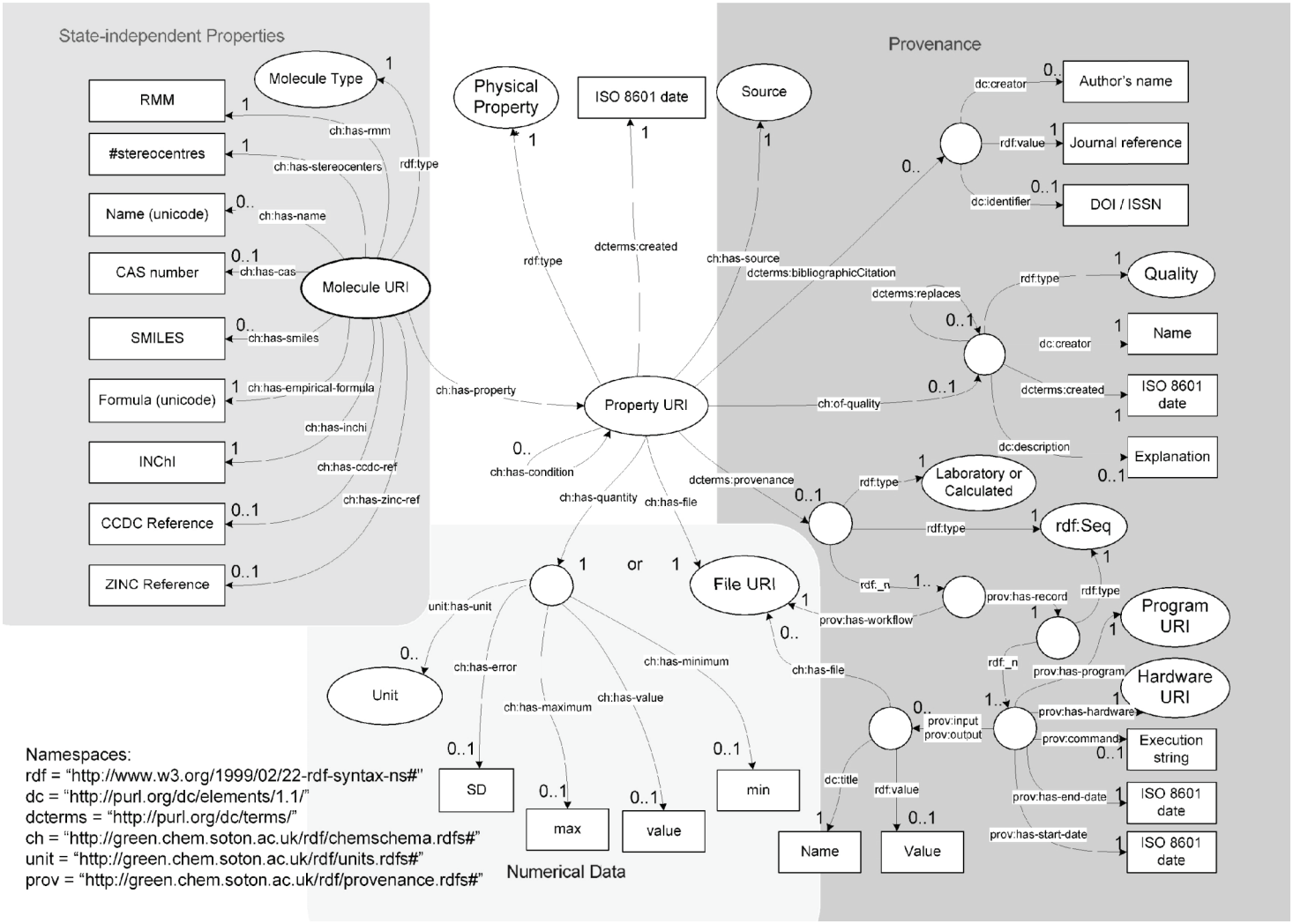

Fig. 1. The schema for the CombeChem datagrid, based around chemical properties. Objects are marked as ellipses, the arrows show how predicates link objects together, and rectangles are literal values. Literals are the actual data in text form and are the values that we need to store, while the rest are simply the data that explains what the literals are. Any combination of ellipse or rectangle and the connecting arrow represents an RDF triple. The tables contain classes of object, such as particular physical properties being defined as melting points. These tables help to explain possibilities that cannot be included in the diagram directly.

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 structure. Associating 3D structures with methods allows several structures to be given for a molecule, 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. This implied ontology is used to provide the meaning for these files without further explicit definition within the store.

Yet another division that was considered related to the phase of a property. For example, one may have a density of substance in any of common states of matter, or computed structures may be created in 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 was not implemented here. This information is also implicit in the other data we would like to store about properties. Properties in which phase is important should be stored alongside the conditions that control the phase, such as temperature. If we do not already know the melting or boiling point of this compound, then we have no way of deciding which phase to put it in, and hence we would be forced to leave this facet as unknown. We have therefore left this issue for further consideration in the future. However the benefits of all such classifications on ease of filtering search results are important in constructing useful chemical queries.

With this new approach applied to data capture and storage, it is possible to filter data based on author, data source, method, accuracy, conditions and molecular properties like relative molecular mass. None of these filterable properties are particularly new but together they far exceed the scope of presently available products. Even more usefully, we can also

isolate data that is unexpected and examine the supplementary information for possible reasons for its abnormality. If it should appear anomalous and that the original data proved incorrect, we can mark it as untrustworthy so that others will know not to place too much faith in it. 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. We maintain a trail of precedence 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 provide a manageable way to make scientific units machine-parseable. RDF is used to create a network of units and quantities that can be effortlessly 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. The result of this is a system for which is easier to write software. 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.

C. Choosing RDF storage technology

The searching and storage of RDF triples in bulk is a difficult prospect. In CombeChem we have experimented with three RDF triplestores:

- *Jena* is a Java framework for building Semantic Web applications, available from HP Labs [13] as open source software under a BSD license. Jena implements APIs for RDF and OWL, and using JDBC can couple with existing RDBMS such as MySQL or PostgreSQL or store triples in system memory. It offers RDFS reasoning over in-memory stores, and RDQL queries [14].
- *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 [15]. It also supports RDFS reasoning, can communicate using a variation of the 'Open Knowledge Base Connectivity' (OKBC) protocol and answer RDQL queries.
- *Kowari* is a Java based triplestore available under the Mozilla Public License from Tucana Technologies [16]. It does not rely on an external RDBMS to provide the actual store and supports queries in a query language called iTQL.

Other RDF storage systems, such as Sesame were not considered at this stage as they were known to not scale to the level of data required by this project. We adopted 3store because it has good scaling properties. Additionally it is easily batch or perl scriptable, supports RDFS scalably, and it can use RDBMS tools for maintenance of data (e.g. backups and migration) as all application state is held in the database, in contrast to Kowari.

D. Using 3store

3store uses an independent database schema for flexibility, and is based around a three level architecture. The top level (an Apache server module) passes an RDQL query to the middle layer (a C library), which compiles the query down to SQL and executes it. MySQL provides the low-level indexing, query execution and persistent storage, which constitutes the bottom layer. This design allows the system to perform query optimisations at each level of abstraction and the final query is translated into one SQL query that can executed by the database engine, in a conventional RDBMS manner, rather than as fragmented queries. There is a trade-off between complexity at query time and store time which is optimized by 3store. This design brings the execution time of typical RDQL queries down to a few milliseconds, and allows for RDF(S) data files to be asserted at a rate of around 1000 triples/second on a commodity x86 based server, even with large knowledge bases.

3store provides utilities that allow interface with the triplestore:

import Takes RDF files, parses them and inserts the triples into the database.

rebuild taxonomy Infers triples based on the schemas loaded via *import*. This is run if the schema is changed, or after any major import.

optimize Rebuilds the mySQL indices to accelerate query speed.

info Supplies summary information about the database, including numbers of files and triples present.

setup Performs the necessary interactions with mySQL for an operational triplestore.

rdql The means of querying the triplestore. It accepts RDQL statements from a number of sources including Web page, command line and perl module.

The RDQL implementation of the version of 3store used in CombeChem is restricted (reduced value constraint expressivity) but also enhanced with graph level provenance support. It can match triple patterns up to a limit of approximately 16 at once due to the underlying mySQL server. This does not pose a problem for most uses, but was problematic when querying our rather deep data structure. 3store can access the full power of mySQL regular expressions and hence provides a very powerful system to perform inexact text matching.

3store is strongly attached to an underlying layer of RDF files – RDF exists in documents so that it can be web addressable. Nothing can change within the database without changing the original file from which the triples came. This demands that we maintain a very large collection of RDF files and administer those files directly. This is contrary to normal database operation where no such files are required. Manipulating text files is both slow and difficult as well as being irreversible, but the benefit is that the RDF files are self-contained packets of information that are completely portable to any triplestore or RDF viewer, and present an excellent

medium for sharing of data. In this context, the triplestore is a rapid access index for all of these files and their content.

The basic unit of RDF we are using is that of one file per molecule. This is sensible and manageable where individual molecules are concerned, but with millions of files it is appropriate to optimise their size and distribution across the filesystem. Fortunately the triplestore keeps the locations of these files internally and thus allows you to trace back to them if needed.

VI. Results

A simple web interface has been constructed that allows all information about a particular molecule to be returned in its structured form. 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 the 80 million triple knowledge base, this exploration of the RDF remained brisk enough for realistic use. The number of different indices available make 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.

Using the datagrid we can rapidly select subsets of scientific data that was recorded on particular dates using particular methods, differentiating between experimental and predicted results such that we know where each and every data point came from as well as how reliable it is. Where one result has been derived from another, the datagrid can tell us which results created the present entry and thus we can deduce the knock-on effects of a correction on the old data. With appropriate workflow enactment it will be straightforward to re-run whatever processes were used to produce the present entry and obtain a new answer.

The ability to add new properties (either measured or calculated) easily to the semantic structure, and then be able to integrate them with the existing data, irrespective of where the underling data is actually stored, is one of the major gains provided by the semantic datagrid. Chemical model building in for example QSAR (quantitative structure activity relations) or QSPR (quantitative structure property relationship) used to predict and screen possible drug molecules, has now reached a level of complexity that a wide range of chemical descriptors is needed to provide sufficient flexibility 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 – thus the need for the datagid. 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 molecules 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 information as well as the descriptor values.

RDF has been shown to be an effective method for capturing highly detailed chemical data and allows it to be indexed in a persistent triplestore such that it can be searched and mined in useful ways. The triplestore has now reached a state of minimal operability. Further addition of chemical properties is an ongoing process. We are now beginning to develop automated calculations using the many available structures, and to store the results alongside all the details of the computations that produced them. Beyond that we can achieve high-throughput data processing and begin to develop new models based on those computations.

We have managed to feed approximately 80 million triples into 3store. Queries remain responsive, but data import performance is degrading, i.e. reassertions of the RDF schema are taking a long time. Write performance on large stores is known to be a challenging issue and we can see how a single large triplestore with frequent insertions would be unable to cope with potential demand. 80 million triples equate to a reasonably sized chemical dataset, but could easily be doubled or trebled when populating with computed properties. Hence we are now contemplating alternative ways of partitioning and maintaining the triples across multiple stores.

VII. Conclusion

The CombeChem Semantic Datagrid has demonstrated how an RDF triplestore can be used to provide enhanced recording, storage and retrieval of scientific data, in a flexible fashion. The triplestores contain the rich metadata that describes the relationships within the scientific information, and the data that is described may be held in a variety of existing stores. Since the metadata is machine-processable, it provides the necessary basis for sophisticated querying and for automation in information processing, which could, for example, include curation [17]. The analysis of the complex data and provenance information needed for chemical information provides valuable lesions for representation and handling of the necessary level of detail involved with data in other sciences.

The Semantic Web is an ambitious goal 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, banishment of the proprietary file format, but there are parties who do not want to make data more easily available to their competitors. However, we have demonstrated the value of adopting this approach on the scale of our project. There is another side to RDF, not often trumpetted by web developers, and that is its use as a local storage and reference system. Not all data needs to be made accessible on the Web, and intellectual property issues may prevent such publication, but the flexibility of the RDF triple model allows it to be applied with several key advantages over conventional approaches.

REFERENCES

[1] J.G.Frey, M.Bradley, J.W.Essex, M.B.Hursthouse, S.M.Lewis, M.M.Luck, L.Moreau, D.De Roure, M.Surridge and A.Welsh, 'Combinatorial Chemistry and the Grid', published in 'Grid Computing: Making the Global Infrastructure a Reality', edited by F.Berman, G.Fox and T.Hey, Wiley (2003).

[2] Tony Hey and Anne E. Trefethen, 'The UK e-Science Core Programme and the Grid', Future Generation Computer Systems 18 (2002) 1017-1031.

[3] J. G. Frey, D. De Roure, L. A. Carr, Publication at Source: Scientific Communication from a Publication Web to a Data Grid, Euroweb 2002 Conference, The Web and the Grid: From e-Science to e-Business. British Computer Society, 2002.

[4] T.Berners-Lee, J.Hendler and O.Lassila, 'The Semantic Web', Scientific American, Vol. 284, No. 5, pp 34-43, May 2001.

[5] D.De Roure, N.R.Jennings and N.R.Shadbolt, 'Research Agenda for the Semantic Grid: A Future e-Science Infrastructure', National e-Science Centre, Edinburgh, U.K., UKeS-2002-02, Dec. 2001. Also see De Roure, D. Jennings, N.R. Shadbolt, N.R. The Semantic Grid: Past, Present, and Future, Proceedings of the IEEE, Volume 93, Issue 3, March 2005, Pages 669-681.

[6] C. A. Goble, D. De Roure, N. R. Shadbolt, and A. A. A. Fernandes, "Enhancing Services and Applications with Knowledge and Semantics," in *The Grid 2: Blueprint for a New Computing Infrastructure*, I. Foster and C. Kesselman, Eds.: Morgan-Kaufmann, 2004, pp. 431-458.

[7] InChI International Chemical Identifier http://www.iupac.org/inchi/

[8] World Wide Web Consortium, Resource Description Framework http://www.w3.org/rdf/, 1997.

[9] G.Hughes, H.Mills, D.De Roure, J.G.Frey, L.Moreau, m.c.schraefel, G.Smith and E.Zaluska, 'The semantic smart laboratory: A system for supporting the chemical e-Scientist', Org. Biomol. Chem. Vol. 2, No. 22, pp3284-3293, 2004.

[10] m.c.schraefel, G.Hughes, H.Mills, G.Smith, T.Payne and J.Frey, 'Breaking the Book: Translating the Chemistry Lab Book to a Pervasive Computing Environment', published in Proceedings of the Conference on Human Factors (CHI), 2004.

[11] P. Murray-Rust, "The World Wide Molecular Matrix - a peer-to-peer XML repository for molecules and properties," presented at EuroWeb2002, Oxford, UK, 2002.

[12] J. D. Myers et al, "A Collaborative Informatics Infrastructure for Multi-scale Science," presented at Challenges of Large Applications in Distributed Environments (CLADE) Workshop, Honolulu, 2004.

[13] Jena – A Semantic Web Framework for Java, http://jena.sourceforge.net/

[14] Andy Seaborne. RDQL - A Query Language for RDF, W3C Member Submission 9 January 2004. http://www.w3.org/Submission/RDQL/

[15] Harris, S and 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, USA.

[16] Kowari Metastore, http://www.kowari.org/

[17] D. De Roure, "On Self-Organization and the Semantic Grid," in IEEE Intelligent Systems, vol. 18, 2003, pp. 77-79.

APPENDIX A– RDF FRAGMENT

```
<rdf:RDF
    xmlns:b = http://green.chem.soton.ac.uk/rdf/chemschema.rdfs#     xmlns:a = "http://www.w3.org/2000/01/rdf-schema#"
    xmlns:d = http://purl.org/dc/elements/1.1/     xmlns:c = "http://purl.org/dc/terms/"
    xmlns:rdf = "http://www.w3.org/1999/02/22-rdf-syntax-ns#">
    <rdf:Description rdf:about = "file:/home/dump/triplestore/rdf/5/3/6/53619a07239f3aae6de6a86ab2264315.rdf">
        <a:type rdf:resource = "http://green.chem.soton.ac.uk/rdf/chemschema.rdfs#OrganicMolecule"/>
        <b:has-inchi>/C18H22N2O2S/c1-5-18(16-9-8-14(3)15(4)12-16)19-20-23(21,22)17-10-6-13(2)7-11-17/h6-12,20H,5H2,1-
4H3/b19-18+</b:has-inchi>
        <b:has-simple-inchi>C18H22N2O2S/c1-5-18(16-9-8-14(3)15(4)12-16)19-20-23(21,22)17-10-6-13(2)7-11-17/h6-12,20H,5H2,1-
4H3</b:has-simple-inchi>
        <b:has-empirical-formula>C18H22N2O2S</b:has-empirical-formula>
        <b:has-stereocentres>0</b:has-stereocentres>
        <b:has-property>
            <rdf:Description rdf:about = "uri://green.chem.soton.ac.uk/property/339788">
                <a:type rdf:resource = "http://green.chem.soton.ac.uk/rdf/chemschema.rdfs#Structure"/>
                <b:has-source rdf:resource = "http://green.chem.soton.ac.uk/rdf/sources.rdfs#NCI"/>
                <b:of-quality>
                    <rdf:Description>
                        <a:type rdf:resource = "http://green.chem.soton.ac.uk/rdf/chemschema.rdfs#Good"/>
                    </rdf:Description>
                </b:of-quality>
                <c:provenance>
                    <rdf:Description>
                        <a:type rdf:resource = "http://www.w3.org/2000/01/rdf-schema#Seq"/>
                        <a:type rdf:resource = "http://green.chem.soton.ac.uk/rdf/chemschema.rdfs#Calculated"/>
                        <a:_1>
                            <rdf:Description>
                                <b:has-description>http://green.chem.soton.ac.uk/methods/ncicorina.htm</b:has-description>
                            </rdf:Description>
                        </a:_1>
                    </rdf:Description> …
```