Proceedings of the UK e-Science All Hands Meeting 2005, ©EPSRC Sept 2005, ISBN 1-904425-53-4

Towards Data Integration for Computational Chemistry.

Philip A Couch, Paul Sherwood, Shoaib Sufi, Ilian T Todorov, Robert J Allan [1]
Peter J Knowles [2]
Richard P Bruin, Martin T Dove [3]
Peter Murray-Rust [4]

1. CCLRC Daresbury Laboratory, Daresbury, Warrington WA4 4AD, UK.
2. School of Chemistry, Cardiff University, Cardiff CF10 3AT, Wales, UK.
3. Dept. of Earth Sciences, University of Cambridge, Downing Street, Cambridge CB2 3EQ, UK.
4. Unilever Centre for Molecular Science Informatics, Department of Chemistry, University of Cambridge, Lensfield Road, Cambridge CB2 1EW, UK.

Abstract.

Increasingly, understanding complex problems in computational chemistry requires the use of several applications as components of computational workflows. Unfortunately, this process is severely hindered by a lack of data interchange standards. The *e*CCP and *e*Minerals UK *e*Science projects are developing a framework to facilitate the interoperability of such applications. Data sets are serialised in a way that conforms to existing XML languages, including the Chemical Markup Language, and collaboratively designed extensions. Various methods are used to express relationships between data sets, based on the XLink standard. Explicit semantics are associated to data sets by mappings to concepts specified in OWL ontologies. These mappings make use of RDF and XPointer standards and enable the users of framework data to work with its logical structure. This provides a number of advantages, including the ability to work with data that conforms to differing data models. Framework tools are under development that will enable the manipulation of such data, being designed with an emphasis on easy integration into the current working practices of computational communities, such as the UK's Collaborative Computational Projects.

1. Introduction.

The emergence of Grid technology provides an infrastructure for the interoperability of computational codes. This technology is very timely, since the understanding of many complex scientific problems increasingly requires the use of a number of specialised applications used together. Good examples can be found in biochemistry, such as in the study of enzyme reactions. A detailed treatment is required to understand the active site, but it is not possible to treat the whole structure in such detail, since this could be too computationally expensive. In such cases hybrid methods are often used, such as combined quantum mechanics and molecular mechanics.

Many computational chemistry codes work with their own file format for both the input and output data. This is usually organised for efficient representation of the data sets and relationships between them. Often this data is ASCII to ensure portability, although binary data is also used in conjunction with tools such as HDF [1] and netCDF [2]. The data documents contain a great deal of implicit information often provided through the description and document ordering of the data sets. A detailed understanding of the data format is required to be able to extract this, often critical, implicit information.

The lack of data standards in computational chemistry is a significant hindrance to the use of such codes in workflows comprising multiple applications. Although it may be possible to construct a framework for the specification and execution of computational workflows, certain restrictions apply. Where data exchange is necessary, the components must either share the same data format or use an existing converter. The use of converters adds unnecessary nodes into the workflow,

tends to be a rather error prone process and does not scale favourably with the number of applications involved.

The UK *e*Science funded *e*Minerals project [3] is a pilot project concerned with the challenge of using computer simulations performed on multiple lengths and time scales starting from the molecular level to address important environmental issues. Project members are distributed throughout the UK and it is important that the members of this virtual organisation are able to efficiently share their data. This data often forms the basis of new computations going from one level to the next and it is desirable to form workflows from components, which can exchange data without human intervention. The *e*CCP project is investigating data management strategies for the UK's Collaborative Computational Projects (CCPs) [5]. These communities are concerned with the development, maintenance and distribution of computational codes and the promotion of best practice in many areas of science and engineering. The current focus of our work has been on CCP1 (quantum chemistry). It is an important requirement that the standards and tools we develop are easy to integrate into the current working practices of the long-established CCP communities.

The *e*CCP and *e*Minerals projects are together developing a framework to facilitate the data interoperability of computational codes, considering W3C and semantic Web community standards and tools. There are a number of important issues that must be addressed when developing such a framework and these are outlined below.

2. Data Models.

2.1. Abstract model

Firstly, a suitable abstract model should be chosen for the data; two choices have been considered. The first is the XML Infoset that imposes a hierarchical view of the data. Data is modelled in terms of components such as elements, attributes and character data. The second is the directed-graph view of RDF; data is modelled in terms of triples (subjects, predicates and objects). There are advantages and disadvantages to both models. Scientific data can be heavily cross-referenced and this produces some difficulty when adopting a purely hierarchical view. In such cases, a lot of referential data becomes necessary to avoid duplication of data sets. The directed graph view solves some of these problems and the current framework can be considered to mix these models.

2.2. Physical model

A further consideration is the specification of the physical data model. XML data can be modelled with schema languages such as RELAX NG, W3C XML Schema or Schematron and XML provides a syntactic standard for its serialisation. There are several standards for the serialisation of RDF data. These include an XML serialisation (RDF/ XML) and others such as N-Triples. Various tools are now available to convert between the different formats (such as the Raptor RDF Parser Toolkit [6]). Due to the relative maturity and wide spread adoption of XML standards and tools over those for RDF, framework data sets are serialised in XML. The data model comprises a number of W3C XML Schema components that model individual data sets. Where possible, these components are taken from existing XML languages. Computational chemistry is fortunate in that the Chemical Markup Language (CML) [7] provides much of what is needed. CML includes a number of XML schemas that specify the representations for data that relate to various chemistry domains. CML has excellent support for the representation of small molecule structures, and is now supported by a number of tools (such a Jmol [8], JChemPaint [9], and Marvin [10]). Where they exist, further representations are adopted from other XML languages. For example, the Visualisation Tool Kit (VTK)[11] has XML representations for properties on grids (regular, rectilinear, irregular). As with the use of CML, the adoption of representations from other XML languages provides benefits through the availability of tools that understand data in this format. The use of VTK markup for grids results in simple visualisation of this data using the VTK APIs.

The *e*CCP project has a Wiki site [4] and e-mail list (http://forge.nesc.ac.uk/pipermail/eccp-data/) that have started to be used for the collaborative design of models that are not available elsewhere (for example, that for Gaussian atomic basis sets).

In some cases, it is a good idea to group concepts into classes and attempt to find a common representation for data that relates to each abstract class. For example, there are many scalar properties that are important in computational chemistry and need to be supported in the phsical data model. Each

scalar property could be considered separately, such as the electron exchange energy or kinetic energy. However, this would lead to a rather large XML schema that would need constant revision as new scalar properties are added. An alternative is to find a common representation for all scalar properties. CML implements this by specifying representations for data that relate to several abstract concepts such as scalar, matrix and array.

Complex relationships exist between computational chemistry data sets and difficulties arise when trying to efficiently represent the data, relationships and semantics simultaneously. Consequently, a modular approach has been adopted with various methods being used to represent relationships. XML nesting is used to a minimum and XLink [12] is used to link data sets and form associations. It also provides semantics for these associations. For example, it can be used to assign atomic basis functions to particular atoms of a molecule, or link a set of calculated properties to a crystal structure.

```
<link id='link1'>

<locator xlink:label='atom1'
xlink:href="#xmlns(cml=http://www.xml-
cml.org/schema/cml2/comp)xpointer(//cml:atom
[@elementType='C'])"/>

<locator xlink:label='basis1'
xlink:href="#xmlns(eccp=http://www.grids.ac.uk
/eccp/ns#)xpointer(//eccp:atomicBasisSet[@id=
'basis1'])"/>

<arc xlink:from='atom1' xlink:to='basis1'/>

</link>
```

Figure 1: *Example of an atomic basis set assignment; link elements in black, locator elements in green, arc elements in red and XPointer expressions in blue.*

The framework provides a vocabulary and syntax for specifying such relationships. The 'link' elements are of XLink extended type. These have child 'locator' elements that are of XLink locator type and identify data sets through the 'xlink:href' attribute. The attribute value is an XPointer [13] expression that is evaluated to resolve data sets. The 'arc' elements are XLink arc type elements and specify directed links between the data sets identified by the 'locator' elements. In Figure 1, links are made between atom data sets and appropriate basis set data sets. In this case, these links represent a basis set assignment. The 'link' elements can also be used to avoid repetition of data. For example, in an XML basis set library, it would be desirable to use components of a 6-31G basis set for that of 6-31G*.

Associations can be specified using 'complex' elements. Here the semantics differ slightly from the XLink standard. There are no XLink arc type child elements; links are bi-directional between data sets identified by the 'locator' elements and the data set that is the association. The XLink semantic attribute 'xlink:role' is used to relate the association to a computational chemistry concept. In the current framework, the value of this attribute is a URI with the prefix 'http://www.grids.ac.uk/eccp/owl-ontologies#'.

```
<complex id='NormalMode1'
xlink:role='http://www.grids.ac.uk/eccp/owl-
ontologies#NormalMode'>

<locator
xlink:href="#xmlns(cml=http://www.xml-
cml.org/schema/cml2/comp)xpointer(//cml:mole
cule[@id='mol1'])"/>

<locator
xlink:href="#xmlns(cml=http://www.xml-
cml.org/schema/cml2/comp)xpointer(//cml:array
[@id='NC1'])"/>

</complex>
```

Figure 2: *Association of normal coordinates and a molecular structure and relation to the concept 'http://www.grids.ac.uk/eccp/owl-ontologies#NormalMode'; complex elements in black, locator elements in green, semantic attribute in red and XPointer expressions in blue.*

The 'link' elements do not require the user to relate links to concepts (e.g. Figure 1), giving some freedom to specify them without providing formal semantics. The 'complex' elements (e.g. Figure 2) are used to specify associations that should be related to concepts and an XLink semantic attribute is used for this purpose. These links and associations are the explicit specification of the often implicit relationships in standard code ouputs. There can be a significant number of these and the framework provides a method for concisely codifying them. Figure 3 shows the specification of 100 associations that relate to the concept 'http://www.grids.ac.uk/eccp/owl-ontologies#NormalMode'

```
<complex id='NormalMode||$1||'
xlink:role='http://www.grids.ac.uk/eccp/owl-
ontologies#NormalMode' instances='100'>

<locator
xlink:href="#xmlns(cml=http://www.xml-
cml.org/schema/cml2/comp)xpointer(//cml:mole
cule[@id='mol1'])"/>

<locator
xlink:href="#xmlns(cml=http://www.xml-
cml.org/schema/cml2/comp)xpointer(//cml:array
[@id='NC||(2*$1)-1||'])"/>

</complex>
```

Figure 3: *A shorthand for explicitly specifying 100 associations that relate to the concept 'http://www.grids.ac.uk/eccp/owl-ontologies#NormalMode'; complex elements in black, locator elements in green, XPointer expressions in blue and numerical expressions in red.*

The '$' character indicates a variable and the '||' string marks the extent of an expression. Variables and expressions may appear in 'id' and 'xlink:href' attribute values of 'complex' elements and 'xlink:href' attribute values of 'locator' elements. The value of the 'instances' attribute provides the number of data set associations. For each association, the variables are resolved and the expressions are substituted by their evaluations. The XPointer expressions are then evaluated to locate the data sets. In Figure 3, 100 associations are specified (the value of $1 ranging from 1 to 100). Each association relates a molecular structure to a set of normal coordinates.

In addition to representing computational chemistry data, there is a requirement to be able to represent metadata. This would include data such as the description of the code used to perform a particular calculation along with its version and details of people involved in the computation. The Data Management Group of the CCLRC have worked in close collaboration with various scientific groups (such as ISIS, Rutherford Appleton Lab, UK and SRS, Daresbury Laboratory, UK) to develop an XML schema for the representation of general scientific metadata, the CCLRC Scientific Metadata Model [14]. It is likely that this can be used to fulfil the metadata and provenance requirements of the *e*CCP1 and *e*Minerals projects, and this assertion is currently being tested.

The current approach allows the fine-grained association of metadata with individual data sets using 'link' elements. It provides a way of adding rich metadata to individual components of documents that may be gathered from different sources. For example, for the purpose of a calculation, a crystal structure may be obtained from one database and collated with a basis set from a second. Provenance may be provided for these individual data sets, not just for the collated data as a whole.

2.3. Conceptual model

The previous discussion surrounds the development of a syntactic standard for representing data. This helps with data interoperability, but still leaves some serious problems; these relate to a lack of explicit semantics. XML only implies semantics through element-types and attribute names. In some cases this can be sufficient; tools could be created that 'understand' this implicit information. A common approach in data management involves the creation of a physical data model and from this the automatic generation of tools to manipulate data that conforms to this model. The creation of such a model is often difficult because the structure and vocabulary must efficiently represent the data, relationships and provide clear semantics. In addition, often, users of the tools must have an understanding of the physical data model in order to manipulate the data. The current framework takes an alternative approach.

The *e*CCP framework can work with ontologies that specify the conceptual model for a domain. This requires an understanding of the concepts that are of interest to a particular community. In the current context, examples would include concepts such as *molecule*, *crystal*, *basis set* and *molecular orbital*. The project members have been involved in a number of meetings held to consult with the international computational chemistry community and gather requirements (e.g. Towards a Common Data and Command Representation for Quantum Chemistry [15]; a summary can be found at the *e*CCP1 Twiki site [4]).

Once there is an understanding of the conceptual model, a formal method of specifying this is required. The current approach is to use the Ontology Web Language (OWL) [16] for this purpose. OWL is used to specify classes of entities, their relationships and their properties. The ontology specifies the logical structure of the data. The current ontologies have been created using the Protégé ontology editor and

knowledge acquisition system [17], along with the OWL plugin.

3. Logical-physical Mapping.

With a specification of the logical structure, users can be abstracted away from the physical structure of the data. In order to facilitate this separation, information must be provided to enable a mapping between such structures. Mappings to XML data are currently implemented using RDF and XPointer. RDF is used to associate XPointer expressions with OWL classes and properties. These XPointer expressions locate data sets and data items in XML documents. This provides a mechanism to associate OWL classes with data sets of particular entities and OWL properties with data items, see Figure 4.

```
<rdf:Description
rdf:about="http://www.grids.ac.uk/eccp/owl-
ontologies#Molecule">

<ns0:xlocator
xmlns:ns0="http://www.grids.ac.uk/eccp/ns#"
rdf:datatype="http://www.w3.org/2001/XMLSch
ema#string">
#xmlns(cml=http://www.xml-
cml.org/schema/cml2/comp)xpointer(//cml:mole
cule)
</ns0:xlocator>

</rdf:Description>
```

Figure 4: *Mapping between the OWL class 'http://www.grids.ac.uk/eccp/owl-ontologies#Molecule' and XML data sets identified by the XPointer expression '#xmlns(cml=http://www.xml-cml.org/schema/cml2/comp)xpointer(//cml:molecule)'; OWL classes in red and XPointer expressions in blue.*

The OWL ontologies and RDF mappings can be stored in repositories for later re-use by other communities and the use of standards to specify the concepts and mappings enable additions/ revisions to be easily made by humans and/ or computers.

Users of the framework data may now work with its logical structure, providing many advantages. Firstly, and most obviously, it is simpler to work with the logical structure. The user is likely to have an understanding of the conceptual model and will not need to be concerned with the details of the data serialisation. The mappings can link concepts with data sets and data items that conform to differing data models, providing some syntactic interoperability. It is a long-term vision that interoperability may also be achieved at the conceptual level though the specification of more complex OWL class and property relationships and the use of OWL reasoners. Some of the data sets will be related to abstract concepts, such as scalar, array and matrix. For such data sets, further semantics are provided via links to XML dictionary entries. The CML data model allows these to be specified by 'dictRef' attribute values.
These dictionaries should be created and maintained by individual communities. The *e*Minerals project already maintains a number of dictionaries that relate to individual computational codes.

4. eCCP Framework Tools.

A library is being developed that allows simple manipulation of framework data. It is writen in C and wrappers will be generated for a range of other languages (including Java, Fortran and Python). The library is designed to be easy to integrate into existing computational projects and has few dependencies (it is built on top of Gnome's libxml2 [18]). Applications that wish to use framework data can work with the appropriate wrapper functions. Applications make library calls to load XML data documents. Functions are passed URIs that locate the data documents on the local file system or at remote sites (transfer by HTTP is supported). Library calls are then made to load supplementary documents. These include RDF/ XML OWL ontologies and logical-physical mappings. The XML data documents can specify the location of supplementary documents that should be loaded and this is done via 'semantics' elements.

Framework data consists of data sets serialised in XML with links and associations specified via 'link' and 'complex' elements respectively. In addition, ontology and mapping constructs can be expressed in the data documents via 'semantics' elements. In this manner, the documents are able to modify the way in which they are interpreted by the parsing library. This adds a great deal of flexibility, since it allows the creator of the data to specify additions/ revisions to both the ontology and mappings. Priority is given to specifications in the data document, followed by those in documents identified by 'semantics' elements, and lastly to those found in supplementary documents loaded by library calls.

Queries are made against the logical structure of the data. Library calls can be made to locate data sets and return data items that relate to

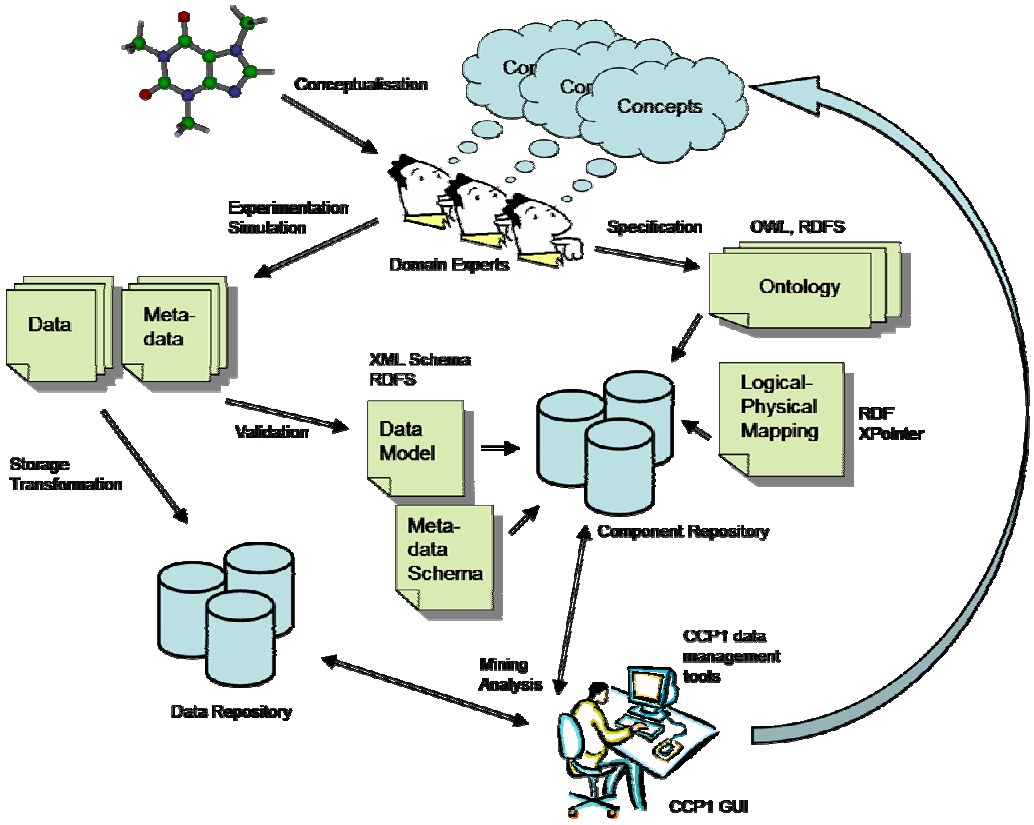

Figure 5: *A data-centric view of the management framework.*

specific OWL classes and properties. The library will operate with no supplementary documents available for a particular data document (where no ontology or mappings are specified). In this case, a number of assumptions are made about the relationship between the logical and physical structures. Users can search for associations of data sets specified by 'complex' elements. For example, a query may request all data sets that relate to the concept 'http://www.grids.ac.uk/eccp/owl-ontologies#NormalMode' and associations of molecular structures and normal coordinates will be returned. Since the links specified by 'complex' elements are bi-directional between the association and the data sets from which the association is comprised, the user may move from a data set to any association which it comprises. For example, a user may request all data sets that relate to molecules and then find all the normal modes of these molecules. The 'link' elements do not function in this way. These elements specify links between data sets and no semantics are defined for the association. Users may navigate from a data set to those to which it is linked. For example, a particular atom data set can be selected and the atomic basis set data set that is linked to this can be found. It is for individual communities to decide how particular data sets should be related.

Default ontologies and mappings will be available for use with the library (for computational chemistry). The user of framework data need not be concerned with their modification, unless support is required for concepts that are not covered or mappings are required to data that does not conform to the standard data model. It is not the intention that users be required to understand the development of the ontologies or mappings. To query the data, the user need only understand its logical structure and the library calls required to load the default ontologies and mappings.

Figure 5 shows a data-centric view of the framework.

5. Conclusions and Future Developments.

This paper has described a standards-based approach to facilitating the data interoperability of computational codes. A framework is under development that supports the representation of computational chemistry data sets in XML. W3C standards, such as XLink and XPointer, are adopted to explictly specify the relationships between data sets. Links are also made between data sets and concepts specified in OWL ontologies. In

addition to providing formal semantics, these allow an abstraction away from the physical format of the data and allow the user to work with the logical structure of the data.

The framework is not restricted to use with computational chemistry data and the tools and principles could be applied to other domains. The library is still under heavy development and additions/ modifications are likely to be made to both the interface and implementation. Wrappers however exist for Python (generated by SWIG [19]) and ones for Fortran are being developed. The current ontologies are class and property definitions and their use is as a restricted vocabulary for querying.

At this stage, the framework can not be used to create the data documents. The current emphasis has been placed on the querying of existing documents, identified by the projects as a key area of difficulty. Applications could write the data documents directly, however there are a number of mature XML tools that can help; these include libxml2 and Xerces. Computational scientists are often involved in the development of applications that have been written in Fortran. This tends to result from the use of 'heritage' libraries that have been highly optimised over a long period of time. Unfortunately, there are very few native Fortran XML tools. The *e*Minerals project has adapted several computational codes such that they output data in a format that conforms to the Chemical Markup Language schema. These codes include Siesta, GULP and DL_POLY. Part of this development has lead to the creation of a native Fortran 90 library that can be used by other developers to create CML output. This has been integrated into xmlf90, a general XML library for Fortran [20]. Library functions include the ability to ensure that the XML is well formed and validates against the CML schema. The *e*CCP project is investigating the integration of the data management tools described with key CCP1 codes such as GAMESS-UK and Molpro, along with tools for visualisation such as the CCP1 GUI.

The logical-physical mappings not only allow an abstraction away from the format of XML data, but also in principle data in any format. This could include binary data stored using technologies such as HDF or netCDF. This is important, since it is not practical to represent all scientific data in XML. The volume of data can be large and the expense incurred in both storing and manipulating the data would be too great. For this reason, the project is considering the extension of the framework to work with non-XML data. The library supports querying of data on the local file system or at remote sites that allow access via HTTP. The *e*Minerals project have chosen to use the Strorage Resource Broker (SRB) [21] to store all of their data. The SRB is middleware that provides a uniform interface to data stored using a range of different technologies and in a range of physical locations. Future developments will allow users of the *e*CCP tools to access data stored in SRB repositories.

6. References.

1. Hierarchical Data Format (HDF), http://hdf.ncsa.uiuc.edu/
2. netCDF, http://my.unidata.ucar.edu/content/software/netcdf/index.html
3. *e*Minerals Project, http://www.eminerals.org
4. *e*CCP project, http://www.grids.ac.uk/eccp
5. Collaborative Computational Projects, http://www.ccp.ac.uk/
6. Raptor RDF parser toolkit, http://librdf.org/raptor/
7. CML: The Chemical Markup Language, http://cml.sourceforge.net
8. JMol, http://jmol.sourceforge.net
9. JChemPaint, http://jchempaint.sourceforge.net
10. Marvin, http://www.chemaxon.com/marvin/
11. VTK: The Visualisation ToolKit, (http://public.kitware.com/VTK/
12. XML Linking Language (XLink) Version 1.0, http://www.w3.org/TR/xlink/
13. XPointer Framework, http://www.w3.org/TR/2003/REC-xptr-framework-20030325/
14. The CCLRC Scientific Metadata Model, http://www-dienst.rl.ac.uk/library/2002/tr/dltr-2002001.pdf
15. Towards a Common Data and Command Representation for Quantum Chemistry, http://www.nesc.ac.uk/esi/events/394/
16. OWL: Ontology Web Language, http://www.w3.org/2004/OWL/
17. Protégé, http://protege.stanford.edu/
18. libxml2, http://xmlsoft.org/
19. SWIG: Simplified Wrapper and Interface Generator, http://www.swig.org/
20. xmlf90, http://fisica.ehu.es/ag/xml
21. SRB: Storage Resource Broker, (San Diego Supercomputer Centre) http://www.sdsc.edu/srb/