# Integrating Chemistry Scholarship with Web Architectures, Grid Computing and Semantic Web

Capstone submitted (May 2010) to the Faculty of the School of Informatics and Computing, Indiana University, Bloomington, in partial fulfillment for the requirements of the degree.

***RAMESWARA SASHI KIRAN CHALLA***

Master of Science

In

CHEMINFORMATICS

Primary Advisor: Dr. David J Wild

Accepted by the faculty of the School of Informatics and Computing, Indiana University, in partial fulfillment of the requirements for the degree Master of Science.

Master of Science Candidate: ____________________________________________

Capstone Instructor: ____________________________________________________

Advisor (s): ___________________________________________________________

## Final Project Approval

| Person | Signature | Date |
|---|---|---|
| Primary Advisor | | |
| Secondary Advisor | | |
| Capstone Instructor | | |
| Program Director | | |
| Graduate Records Administrator | | |
| Associate Dean | | |

Month and Year ____________________

# Contents

## List of Figures:

## Abstract

A chemist given a compound would be interested in knowing the experiments performed using the compound, journals containing the compound and also molecular properties of the compound. Currently this experimental data, bibliographic data, molecular properties data is available in different databases. If there is a way to integrate this data, it would enhance the chemist's knowledge about a given compound. ORE is the model proposed by Digital library community to aggregate resources on the web. ORECHEM is a Research project funded by Microsoft Research and it aims at this integration of experimental, bibliographical and molecular properties data. As a contribution to this ORECHEM project, a workflow that facilitates the enhancement of the molecular structural properties information was developed. REST services, Grid Computing, and Semantic Web technologies have been used as part of this pipeline. Individual REST services were developed to (i) fetch ORE ATOM XML feeds, (ii)extract Crystallographically obtained 3D coordinates in CML format, (iii) generate Gaussian input files and submit Gaussian jobs to Gaussian on HPC Clusters (TeraGrid here), (iv) convert ORE ATOM XML and Gaussian output to Triples and store them into a Triplestore.

## Introduction and Background

Semantic Web is a development of the World Wide Web in which semantics are added to the information or data on the web to enable machines to make assertions between the data. RDF is the data model used to represent the data in Semantic Web. An RDF triple is subject, predicate and object statement (McBride, 2004). The subject denotes the resource, and the predicate denotes traits or aspects of the resource and expresses a relationship between the subject and the object. Subject, predicate and object are represented as URIs. Objects can even be literals. The predicate is usually defined by Ontology.

Moiety in Chemistry is one half of a molecule. A crystal structure usually will have two or more moieties. GAUSSIAN 03(Gaussian) is a general purpose ab-initio electronic structure package that is capable of computing energies, geometries, vibration frequencies, transition states, reaction paths, excited states, and a variety of properties based on various uncorrelated and correlated wave functions (Gaussian03 TeraGrid). There are various basis set functions (derived from Quantum Chemistry) as options in this software and one needs to specify the basis set function to be applied for a particular compound, in the Gaussian input file (Young).

On the World Wide Web, a Resource represents an item of interest, and every resource has a URI. A Resource might have information similar to another. Thus there can be links between the resources. OAI-ORE (Open Archives initiative – Object Reuse and Exchange) (Lagoze, 2008) is an initiative in the Digital Library community which defines standards for aggregating resources on the Web. According to the ORE model an URI is assigned to an aggregation of Resources. Following the Linked Data guidelines a new Resource is introduced to make information about the Aggregation available. This new Resource is called Resource Map and it has a URI and it expresses which aggregation it describes. A Resource Map is implemented in either ATOM XML or RDF/XML, n3, turtle formats.

It is based on this ORE model that the ORECHEM (Microsoft Research OreChem Project) project originated. ORE model applied to Chemistry Scholarship data is the idea behind the ORECHEM project. ORECHEM project is a project funded by Microsoft Research and it's collaboration between Chemistry Scholars and information scientists to develop and deploy the infrastructure, services, and applications to enable new models for research and dissemination of scholarly materials in the chemistry community. Three universities from USA and two from UK are part of this project.

Penn-State University, using Chem-XSeer (ChemXSeer) extracts the bibliographic metadata from the Journal articles. University of Southampton has a huge archive of Crystals called eCrystals (eCrystals), and their experimental data. Thus they contribute the experimental data which is expressed using Experimental Ontology. University of Cambridge has a process called Crystal Eye Process (Crystal Eye) as part of which, CIF files (IUCr's CIF files) are converted into CML (Chemistry Markup Language (CML Wiki)) and even moieties in a particular crystal are extracted and expressed in CML format. These moiety files have their 3D coordinates and Indiana University's role is to run Gaussian software on these moieties on a High Performance Computing clusters. Gaussian software has the capability to compute stable geometries, transition states, latent heats of fusion and other structural properties. Thus as part of this project, the experimental data, the bibliographic data and the molecular properties information is integrated and stored as triples in a Triplestore. This Triplestore is to be hosted on Azure cloud and any Chemist would be able to perform SPARQL queries on the data in the Triplestore.

## Data, Dataflow/Workflow and Objective

As mentioned above the ORE Resource Maps are usually implemented in ATOM XML, RDF/XML. An obsolete ORE ATOM XML feed was provided by the University of Cambridge. ATOM feed had several entries. Each entry was about a Resource Map. Initial atom tags gave information about the author of the Resource Map, its publishing date, copyrights, etc. Then there was a section called <oreatom:triples> (Lagoze C. V., 2008)where all the aggregates related to the Resource Map were described using <ore:aggregates> tags. It is in this <oreatom:triples> section that other Ontologies could be integrated. For example the sample ATOM feed presented in the Figures section [Figure1]s has <chemdomain:hasidentifier> tag and this is based on the Chem-Axiom Ontology developed by the University of Cambridge. Similary the Experiments, Computational process information could be put into this section as triples, based on the Experimental Ontology and Computational Process Ontology developed by the University of Southampton.

Initially, InChI strings were extracted from this obsolete feed and their 3D coordinates were either obtained from PubChem or using Open Eye Babel, and these 3D coordinates were submitted to Gaussian. Later as the project evolved, it was decided to extract in particular the "cml.xml" files available in the <ore:aggregates> section of the <oreatom:triples> section in every entry. However these cml.xml files which had the moieties 3D coordinates were archived as ATOM feeds (ATOM Archive) in a crystal eye repository at University of Cambridge as part of the Crystal Eye Process. Snapshot of the archived ATOM feed is available in the Figures section [Figure 2]. There is a toolkit called Crystal Eye Harvester (Day), which harvests specified number of feeds from Crystal Eye repository into a local directory specified. A RESTful service was wrapped around this tool.

Initial conversions from CML to Gaussian Input format was done using 'Jumbo-Converters' toolkit developed by University of Cambridge. Later this was replaced with 'Semantic Comp Chem' toolkit (Peter Murray Rust) developed by the same group. Sample Gaussian Input is available in the Figures section [Figure 3]. These input files were submitted to Gaussian and outputs were collected. The Gaussian output was to be transformed into triples and this part was not implemented at the time of completion of this Capstone project.

The workflow had two components. One component dealt with the converting Gaussian output obtained by running Gaussian on Crystal Eye Moieties CML files, into triples and storing them into the Triplestore. Another component dealt with the conversion of the ORE ATOM feeds into triples using SAXON-XSLT transformation (Saxon HE Documentation) (Saxon HE Download) or GRDDL transformation (GRDDL Primer) and storing the triples into a Triplestore. The workflow depicting both these components is given in the Figure 6.

The objective of this project had been to develop REST services that did each of the steps in the two components mentioned above. REST services were developed to do the following: fetch ORE ATOM Feeds, transform the ORE ATOM feeds into triples(RDF/XML) and put them into Triplestore, extract the moiety feeds in CML format and generate Gaussian Input from those, submit jobs to HPC clusters. And a workflow was built, based on these REST services.

## REST Services

Since the ORECHEM project deals with aggregating data on the Web and since REST (Fielding) is the way Web already works, REST architecture was chosen. A REST service is invoked using a URL and every resource has a URI.

### *Jersey*

Java JAX-RS Jersey API (Jersey 1.2)was used to build REST services. It had several Java annotations like @Path, @GET, @POST, @PUT, @DELETE, @Produces, @QueryParam(), @PathParam() using which it was easy to perform HTTP GET/PUT/POST/DELETE operations and write services that are based on these operations. Also one could have a single class in which one could specify slight variations in the URL structure and have different methods being invoked returning different output formats. That is one could have a single class returning a String Array or JSONArray (JSON) depending on the URL structure using which a resource is invoked. In Fig 4, if one has 'csv' in the URL structure, harvestFeeds method will be invoked. If one has 'json' in the URL structure, harvestFeedsJSON method will be invoked. Jersey even has a separate server and client API.

### *ORECHEM REST Services*

Following are the REST services developed as part of this project. As the project evolved when changes were proposed, old services were not modified but were retained so that they can be used in other projects. gf18 is the server that these services are currently deployed on and maven builds are being created around these services at the time of writing this report.

1) **InChIExtractor :**
   Extracts InChIs from ORE ATOM feed. Using JDOM, JAXP looks for <chemdomain:hasidentifier> tag and gets its values.
   **Input**: ORE ATOM feed URL
   **Output**: A String of InChIs separated by special characters $$$$$.

2) **InChIto3D:**
   Generates 3D coordinates (in CML format) using Open Eye Babel. Writes the given input InChI into a String and does a system call on 'babel'.
   **Input**: InChI string
   **Output**: 3D coordinates in CML format as a String**.**

3) **CML2Gauss**
   Converts CML into Gaussian input format using Cambridge's Jumbo-Converters toolkit. Writes the Gaussian input into a file in a specific directory on gf18.
   **Input**: 3D coordinates CML String returned by InChIto3D service.
   **Output**: Path to the file on gf18.

4) **Atom2RDF**
   Using atom-grddl.xsl stylesheet and Saxon-HE toolkit, transformation from ATOM XML to RDF/XML is performed and the output is written to a file on the gf18.
   **Input**: ORE ATOM feed URL

**Output**: Path (on gf18) of the file containing RDF/XML

5) **RDFIntoVirtuoso**
Using Jack rabbit WEBDAV client API's Put () method, places the triples file into Virtuoso Triplestore.
**Input**: RDF/XML file path returned by the Aom2RDF service.
**Output**: GraphIRI using which SPARQL queries can be done on the SPARQL end-point.

6) **FeedsHarvester :**
Wrapped around a tool called 'Crystal eye harvester'. The moieties directory should to be available in the ROOT folder in the webapps folder in the Apache Tomcat.
**Input:** "harvester = moiety and numofentries = num. of feeds to be fetched.
**Output:** URLs to cml.xml files which have the 3D coordinates.

7) **CML2GaussianSemCompChem:**
Generates Gaussian input files from the CML files using Cambridge group's 'Semantic Comp Chem' toolkit. It performs HTTP GET on the cml.xml file URL given as input, writes the content to temp file, generates Gaussian input format, writes to a file and places it in the apache2 htdocs folder.
**Input**: Single cml.xml file URL
**Output**: URL of the Gaussian Input file

***Testing the Services:***

REST services developed, were tested by running them on the localhost server and then using a browser as client for HTTP GET operations and also Jersey Client API was used to test both HTTP GET and HTTP POST based services. Also WAR files were created and were deployed on the gf18 server and then curl command was written into a shell script and thus the services were tested. Test Cases are being written for these services at the time of this writing. JUnit and Jersey Client API are being used to write the Jersey Test cases. Snapshot of a simple test case is in Figure 5.

## Grid Computing

### *TeraGrid & OGCE Workflow Suite*

Gaussian Input files were submitted to Gaussian running on TeraGrid. MPI version of Gaussian is available on the TeraGrid. TeraGrid is a network of huge computational resources like supercomputers connected to each other by a network of bandwidth around 50GBPS. To submit jobs to these super computers, OGCE Workflow Suite was used. It constitutes tools to wrap command-line applications as light weight web services, and execute and monitor the workflows.

GFAC, XRegistry and XBaya are the three main tools. GFAC allows users to wrap any command-line application or a REST service as a SOAP protocol based web service. XRegistry is the information repository of the workflow suite enabling users to register, search and access application service and workflow deployment descriptions. XBaya is a Java webstart workflow composer. It is used for composing workflows from web services created by the GFAC, and running and monitoring those workflows. This workflow suite facilitates easy submissions and management of jobs submitted to TeraGrid. ORECHEM REST Services were registered in the XRegistry and a workflow was built using XBaya workflow composer. A snapshot of the XBaya workflow composer is in Figure 5. It took around 10mins to 2 hours to run Gaussian, varying with the input files.

## *Dryad and DryadLINQ*

Dryad is a high-performance, general-purpose distributed computing engine that simplifies the task of implementing distributed applications on clusters of computers running a windows operating system. The Dryad Project of Microsoft Research is investigating programming models for writing parallel and distributed programs to scale from a small cluster to a large data-center (Dryad Project). An academic release (Dryad) of the same was available. Initial idea of the project was to run Gaussian jobs on windows clusters using Dryad. A successful attempt was made to use Dryad for doing simple file format conversions using Jumbo Converters toolkit on different nodes in the windows cluster. Since windows version of Gaussian wasn't available as an open source tool, the usage of Dryad was not continued. However if there is any open source replacement to Gaussian, like MPQC (MPQC ), one could use Dryad to submit jobs to different nodes on a cluster and thus make use of high computing resources.

DryadLINQ allows implementing Dryad applications in managed code by using an extended version of the LINQ programming model and API.  LINQ was introduced with Microsoft .NET framework version 3.5. DryadLINQ provider translates the application's LINQ queries into a Dryad job and runs the job as a distributed application on a windows HPC cluster.

First step in implementing Dryad is to partition the input files on to different nodes on the cluster. In this attempt, from the head node, all the input cml.xml files which were to be converted to Gauss Input using Jumbo Converter toolkit were placed into all the nodes available using C# code. A partition data set consisted of two types of files. One was a metadata file with .pt extension and the other was a partition file on every node which has file paths to the input files. The metadata file had the number of nodes, path to the partition file available in it. Thus DryadLINQ reads through every line in this metadata file, and executes the process on every file in the respective node and places the output files into a particular directory on the head node.

## Semantic Web Technologies

A Triplestore is a framework for storing and querying RDF data in Semantic Web Technologies stack (Triplestore wiki). It provides a mechanism for persistent storage and access of RDF graphs. There are several commercial and open source Triplestores available. Open Link Virtuoso Triplestore was recommended by one of the ORECHEM project members and this was chosen for the project.

### *Open Link Virtuoso Triplestore*

This is an ORDBMS extended into a Triplestore (Erling). It has command line loaders: isql utility (interactive sql access to a database) available. SPARQL is similar to SQL and it is meant for querying RDF data. Virtuoso has a support for SPARQL and a webserver (end point) to perform SPARQL queries. It also has a WEBDAV browser using which one can upload RDF data files. For every file uploaded into a Triplestore there is an IRI created.

Every file that is uploaded into Virtuoso is stored in a directory called 'rdf_sink'. And the triples are stored in a table with four columns: GraphID, Subject, Predicate, and Object with IRI_ID, IRI_ID, IRI_ID, and ANY as respective data types (Erling).

Windows and Linux versions of Virtuoso were installed. Linux Installation procedure is documented here (Challa).

## Future Work

To facilitate the process of getting the new ATOM entry as soon as it is published by any of the four universities in the Project, Google's PubSubHubbub (PubSubHubbub) could be used. It is based on a publisher/subscriber protocol in which a publisher is subscribed to a Hub, and also the subscriber is subscribed to the Hub. Hub can be a Google App Engine or an Red Hat Linux Server. As a new entry is published by the publisher, hub gets the new content and it pushes this new content to the subscriber. Thus here in this project, a Hub could be set up at IU, and be subscribed to it. Remaining Universities can be subscribed to this Hub and as they publish their content, IU can get the new ATOM entries and start of the pipeline of ORECHEM REST services and put the data into the Triplestore. Also that part of the pipeline where the Gaussian Output is to be transformed to Triples is to be developed into a REST service.

## Acknowledgements

I would like to thank Dr. Marlon Pierce for his constant guidance, weekly tasks which let me be within the time line and finish the services as required in the project. I also would like to thank Dr. David. J. Wild for introducing me to Marlon. I would like to thank Suresh Marru for helping me in using the OGCE workflow suite, and also for helping integrate ORECHEM REST services and building a workflow out of them. I also would like to thank every member in the OreChem project group: Dr. Carl Lagoze (Cornell University), Dr. Peter Murray Rust, Nick Day, Jim Downing(University of Cambridge), Mark Borkum (University of Southampton), Na Li (Penn State), Alex, Lee Dirks (Microsoft Research). I would like to thank Jaliya Ekanayake and Scott Beason for helping with the DRYAD, DRYADLINQ work. My thanks to PTI members as well all my classmates and family members.

## Appendix

Here are URLs to invoke REST services. Jersey Client API or curl commands on Linux, can be used to invoke these services.

1) **InChIExtractor :**
   http://gf18.ucs.indiana.edu:8146/InChIExtractor/atom/InChIs?feedurl=$atomurl
   Eg:
   http://gf18.ucs.indiana.edu:8146/InChIExtractor/atom/InChIs?feedurl=http://wwmm.ch.cam.ac.uk/crystaleye/feed/orechem/feed.xml

2) **InChIto3D :**
   http://gf18.ucs.indiana.edu:8146/InChIto3D/inchi/3dstructure?InChIstring=InChIinput
   Eg:
   http://gf18.ucs.indiana.edu:8146/InChIto3D/inchi/3dstructure?InChIstring=InChI=1/H3N/h1H3/p+1

3) **CML2Gauss:**
   One needs to POST content to this. Using curl this is how it is done.
   curl -d "$structure" -H "Content-Type:application/xml"
   http://gf18.ucs.indiana.edu:8146/CML2Gauss/cml3d/gaussinput

4) **Atom2RDF:**
   http://gf18.ucs.indiana.edu:8146/Atom2RDF/atomfeed/rdf?feedurl=$atomurl
   Eg:
   http://gf18.ucs.indiana.edu:8146/InChIExtractor/atom/InChIs?feedurl=http://wwmm.ch.cam.ac.uk/crystaleye/feed/orechem/feed.xml

5) **RDFIntoVirtuoso:**
   POST filepath to this service. Using curl:
   curl -d "$rdf" -H "Content-Type:application/xml"
   http://gf18.ucs.indiana.edu:8146/RDFIntoVirtuoso/rdftriples/virtuoso

6) **FeedsHarvester:**
   This can have String Array or JSONArray as output depending on the service invoked.
   **String Array output:**
   http://gf18.ucs.indiana.edu:8146/FeedsHarvester/crystaleye/cml3d/csv?harvester=moiety&numofentries=5

   **JSONArray output:**
   http://gf18.ucs.indiana.edu:8146/FeedsHarvester/crystaleye/cml3d/json?harvester=moiety&numofentries=5

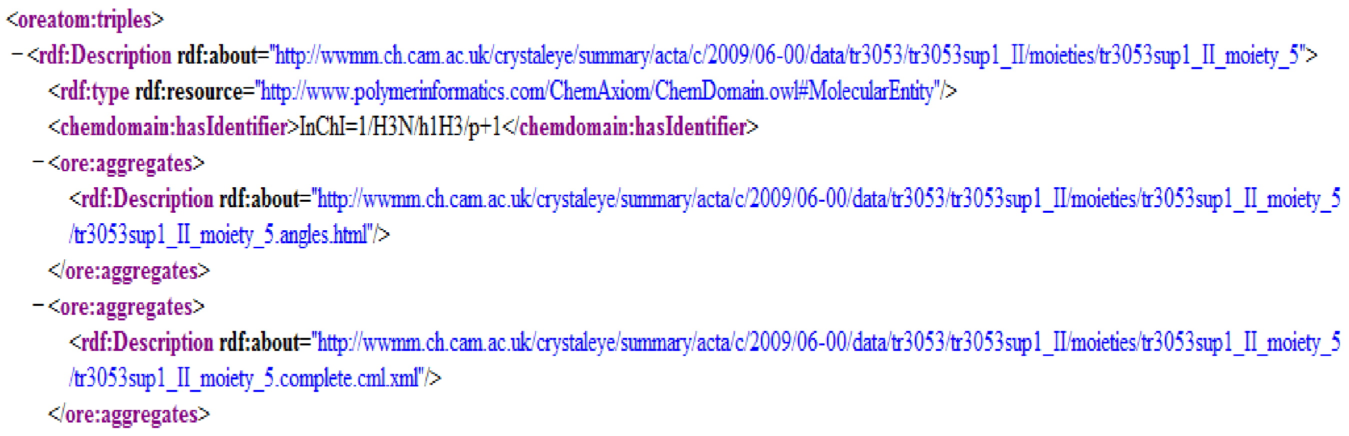
7) **CML2GaussianSemCompChem:**
   Each of the cml.xml file URLs obtained above should be posted to this service.

```
curl -d "$cmlfileURL" -H "Content-type:text/plain"
http://gf18.ucs.indiana.edu:8146/CML2GaussianSemCompChem/gauss/inputgenerator
```

Virtuoso Conductor: http://gf18.ucs.indiana.edu:8890/conductor

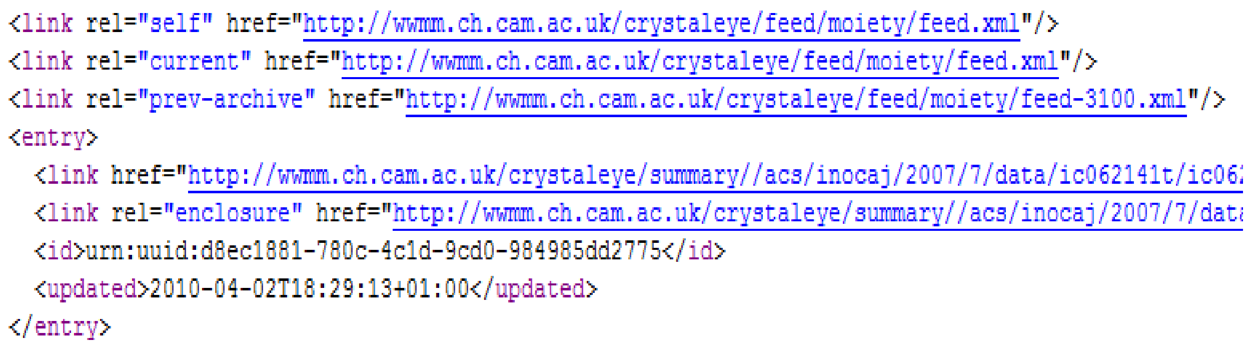
Virtuoso SPARQL endpoint: http://gf18.ucs.indiana.edu:8890/sparql

## Figures



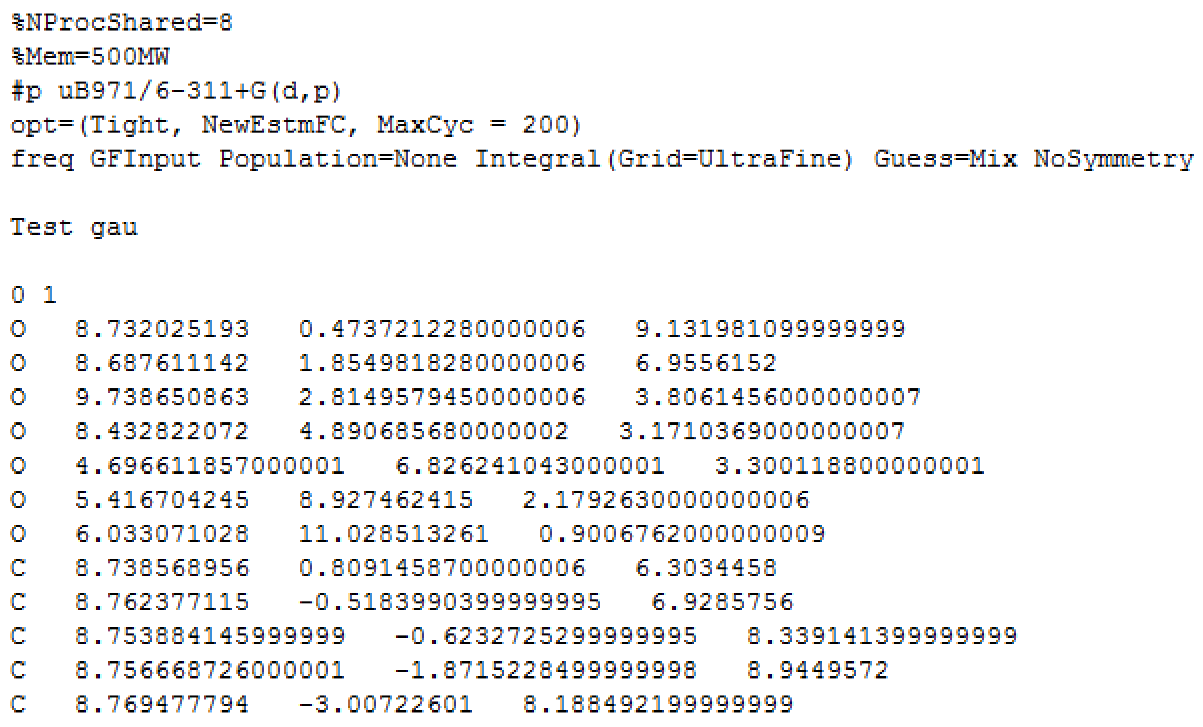
```
<oreatom:triples>
 - <rdf:Description rdf:about="http://wwmm.ch.cam.ac.uk/crystaleye/summary/acta/c/2009/06-00/data/tr3053/tr3053sup1_II/moieties/tr3053sup1_II_moiety_5">
     <rdf:type rdf:resource="http://www.polymerinformatics.com/ChemAxiom/ChemDomain.owl#MolecularEntity"/>
     <chemdomain:hasIdentifier>InChI=1/H3N/h1H3/p+1</chemdomain:hasIdentifier>
   - <ore:aggregates>
       <rdf:Description rdf:about="http://wwmm.ch.cam.ac.uk/crystaleye/summary/acta/c/2009/06-00/data/tr3053/tr3053sup1_II/moieties/tr3053sup1_II_moiety_5
       /tr3053sup1_II_moiety_5.angles.html"/>
     </ore:aggregates>
   - <ore:aggregates>
       <rdf:Description rdf:about="http://wwmm.ch.cam.ac.uk/crystaleye/summary/acta/c/2009/06-00/data/tr3053/tr3053sup1_II/moieties/tr3053sup1_II_moiety_5
       /tr3053sup1_II_moiety_5.complete.cml.xml"/>
     </ore:aggregates>
```

**Figure 1 ORE ATOM based Resource Map**



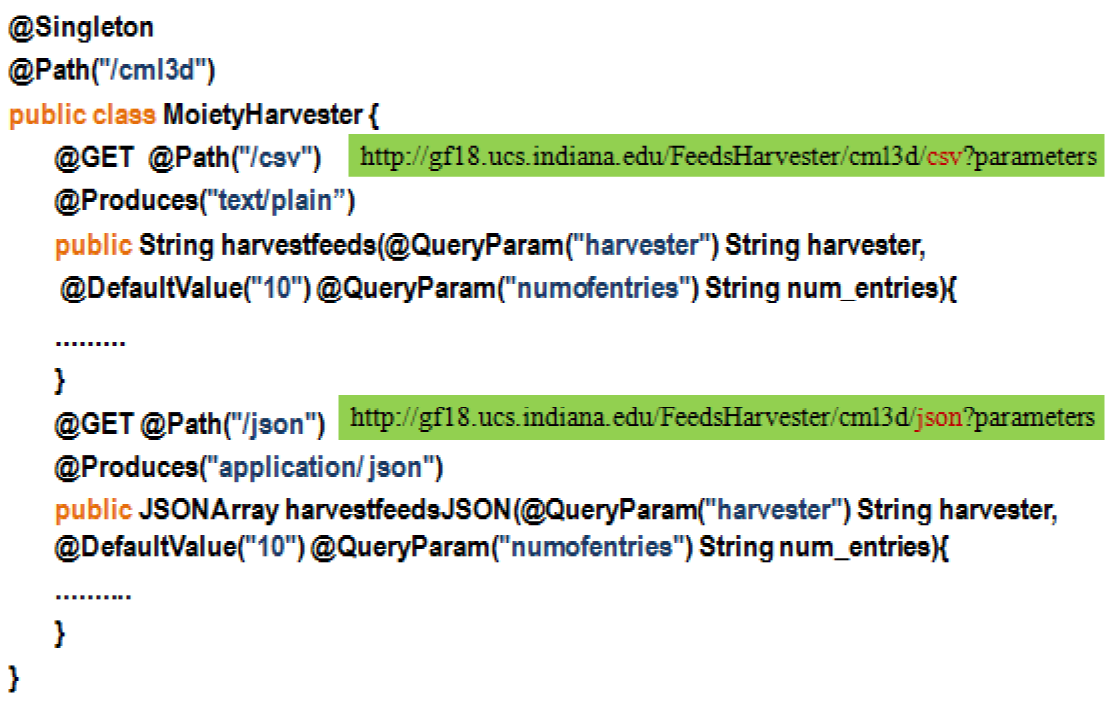
```
<link rel="self" href="http://wwmm.ch.cam.ac.uk/crystaleye/feed/moiety/feed.xml"/>
<link rel="current" href="http://wwmm.ch.cam.ac.uk/crystaleye/feed/moiety/feed.xml"/>
<link rel="prev-archive" href="http://wwmm.ch.cam.ac.uk/crystaleye/feed/moiety/feed-3100.xml"/>
<entry>
  <link href="http://wwmm.ch.cam.ac.uk/crystaleye/summary//acs/inocaj/2007/7/data/ic062141t/ic06
  <link rel="enclosure" href="http://wwmm.ch.cam.ac.uk/crystaleye/summary//acs/inocaj/2007/7/dat
  <id>urn:uuid:d8ec1881-780c-4c1d-9cd0-984985dd2775</id>
  <updated>2010-04-02T18:29:13+01:00</updated>
</entry>
```

**Figure 2 ATOM Archive Moiety Feed**



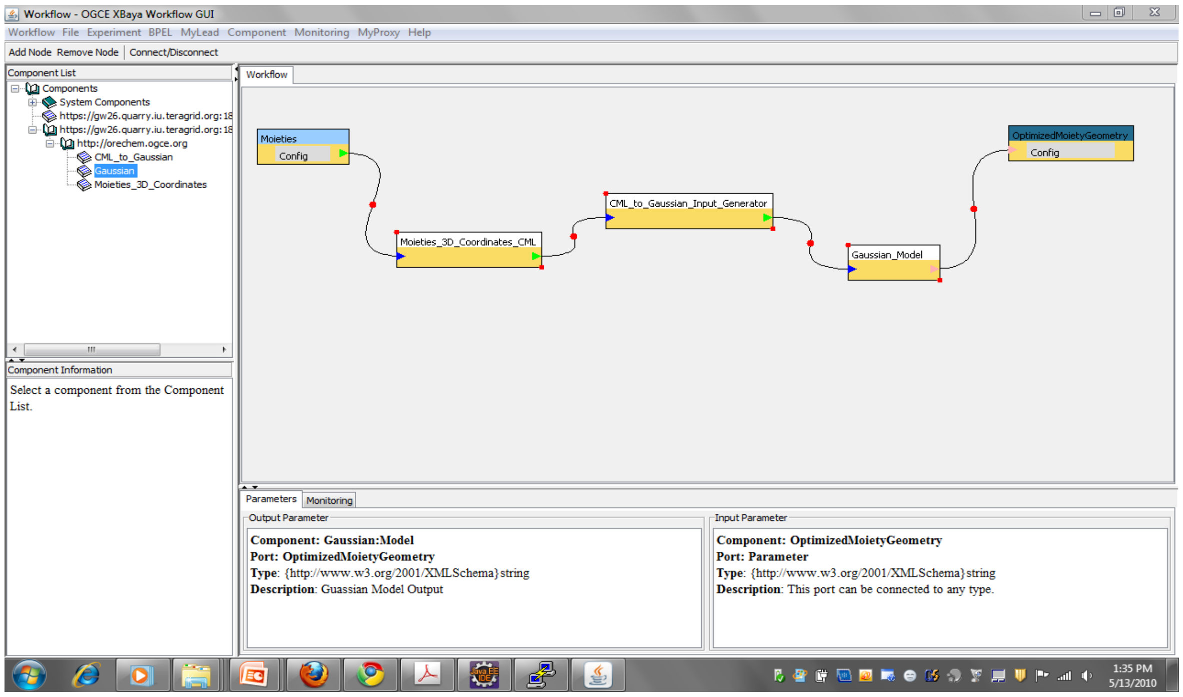
```
%NProcShared=8
%Mem=500MW
#p uB971/6-311+G(d,p)
opt=(Tight, NewEstmFC, MaxCyc = 200)
freq GFInput Population=None Integral(Grid=UltraFine) Guess=Mix NoSymmetry

Test gau

0 1
O   8.732025193   0.4737212280000006   9.131981099999999
O   8.687611142   1.8549818280000006   6.9556152
O   9.738650863   2.8149579450000006   3.8061456000000007
O   8.432822072   4.890685680000002   3.1710369000000007
O   4.696611857000001   6.826241043000001   3.300118800000001
O   5.416704245   8.927462415   2.1792630000000006
O   6.033071028   11.028513261   0.9006762000000009
C   8.738568956   0.8091458700000006   6.3034458
C   8.762377115   -0.5183990399999995   6.9285756
C   8.753884145999999   -0.6232725299999995   8.339141399999999
C   8.756668726000001   -1.8715228499999998   8.9449572
C   8.769477794   -3.00722601   8.188492199999999
```

**Figure 3 Sample Gaussian Input file**



```
@Singleton
@Path("/cml3d")
public class MoietyHarvester {
    @GET  @Path("/csv")    http://gf18.ucs.indiana.edu/FeedsHarvester/cml3d/csv?parameters
    @Produces("text/plain")
    public String harvestfeeds(@QueryParam("harvester") String harvester,
     @DefaultValue("10") @QueryParam("numofentries") String num_entries){
    .........
    }
    @GET @Path("/json")   http://gf18.ucs.indiana.edu/FeedsHarvester/cml3d/json?parameters
    @Produces("application/ json")
    public JSONArray harvestfeedsJSON(@QueryParam("harvester") String harvester,
    @DefaultValue("10") @QueryParam("numofentries") String num_entries){
    ..........
    }
}
```

**Figure 4 Snapshot of REST service written using Jersey. Note the difference in the structure of the URL, difference in output formats & different Method names.**


Workflow - OGCE XBaya Workflow GUI
Workflow File Experiment BPEL MyLead Component Monitoring MyProxy Help
Add Node Remove Node Connect/Disconnect
Component List
Components
System Components
https://gw26.quarry.iu.teragrid.org:18
https://gw26.quarry.iu.teragrid.org:18
http://orechem.ogce.org
CML_to_Gaussian
Gaussian
Moieties_3D_Coordinates
Component Information
Select a component from the Component List.
Workflow
Moieties
Config
Moieties_3D_Coordinates_CML
CML_to_Gaussian_Input_Generator
Gaussian_Model
OptimizedMoietyGeometry
Config
Parameters
Monitoring
1:35 PM
5/13/2010

Output Parameter

**Component: Gaussian:Model**
**Port: OptimizedMoietyGeometry**
**Type**: {http://www.w3.org/2001/XMLSchema}string
**Description**: Guassian Model Output

Input Parameter

**Component: OptimizedMoietyGeometry**
**Port: Parameter**
**Type**: {http://www.w3.org/2001/XMLSchema}string
**Description**: This port can be connected to any type.

**Figure 5 Snapshot of the XBaya workflow Composer showing the usage of ORECHEM REST services**

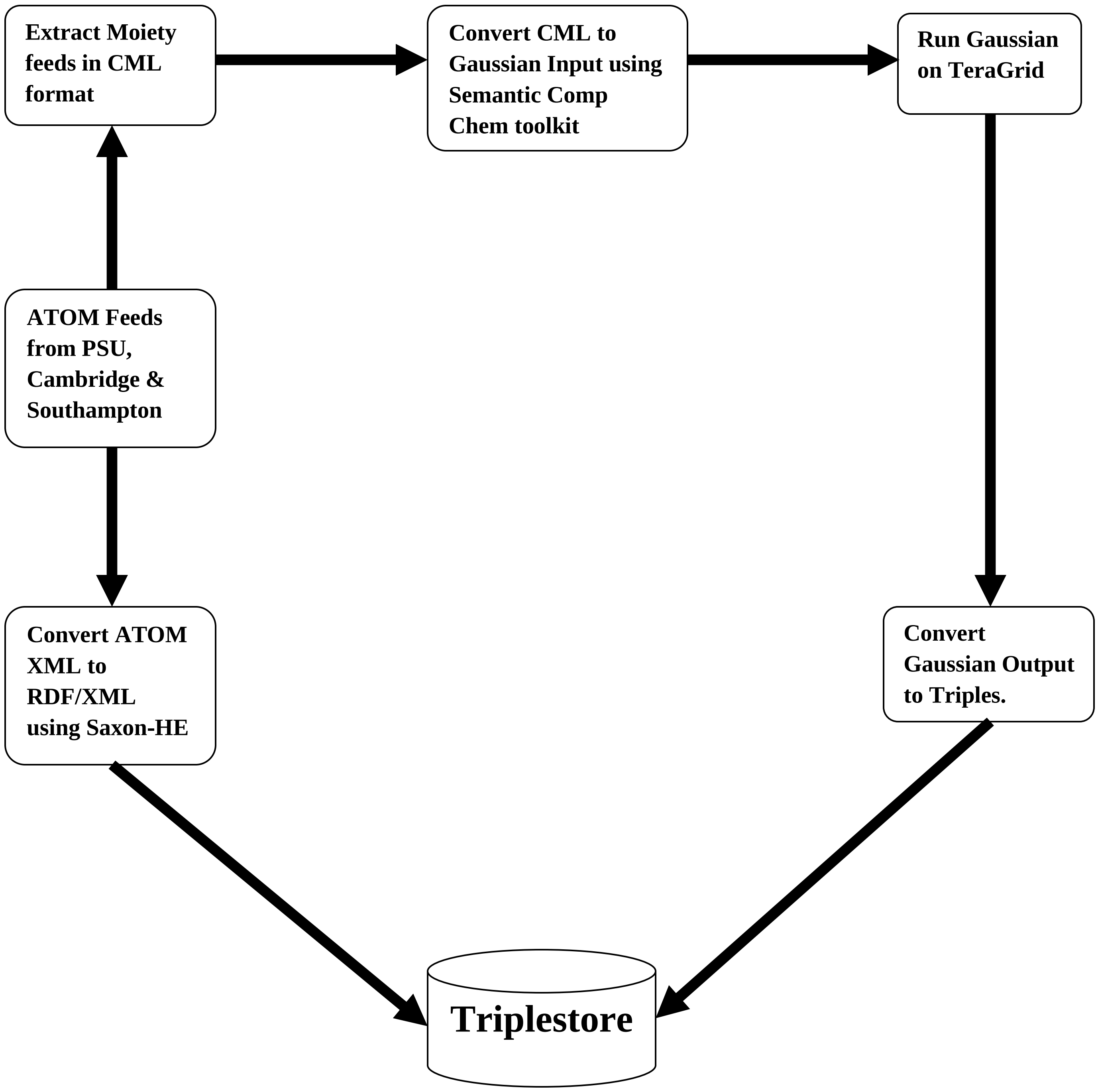

Extract Moiety feeds in CML format
Convert CML to Gaussian Input using Semantic Comp Chem toolkit
Run Gaussian on TeraGrid
ATOM Feeds from PSU, Cambridge & Southampton
Convert ATOM XML to RDF/XML using Saxon-HE
Convert Gaussian Output to Triples.
Triplestore

**Figure 6: Dataflow/Workflow**

## References

*ATOM Archive*. Retrieved from http://tools.ietf.org/html/rfc5005

Challa, S. *Informatics Blog*. Retrieved from http://sashikiranchalla.blogspot.com/2009_11_01_archive.html

*ChemXSeer*. Retrieved from http://chemxseer.ist.psu.edu/

*CML Wiki*. Retrieved from http://en.wikipedia.org/wiki/Chemical_Markup_Language

*Crystal Eye*. Retrieved from http://wwmm.ch.cam.ac.uk/crystaleye/

Day, N. . *Crystal Eye Harvester Bit Bucket Page*. Retrieved from http://bitbucket.org/ned24/crystaleye-harvesters/src/

*Dryad*. Retrieved from http://connect.microsoft.com/site/sitehome.aspx?SiteID=891

*Dryad Project*. Retrieved from http://research.microsoft.com/en-us/projects/dryad/

*eCrystals*. Retrieved from http://ecrystals.chem.soton.ac.uk/

Erling, O. *Implementing a SPARQL compliant RDF Triplestore using SQL- ORDBMS.* Retrieved from http://virtuoso.openlinksw.com/dataspace/dav/wiki/Main/VOSRDFWP

Fielding, R. *REST.* Retrieved from http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm

*Gaussian03 TeraGrid*. Retrieved from http://hpcsoftware.teragrid.org/Software/user/show_asset.php?asset_id=124&view=TeraGrid

*GRDDL Primer*. Retrieved from http://www.w3.org/TR/grddl-primer/

*IUCr's CIF files*. Retrieved from http://www.iucr.org/resources/cif

*Jersey 1.2*. Retrieved from https://jersey.dev.java.net/nonav/documentation/latest/user-guide.html

*JSON*. Retrieved from http://www.json.org/fatfree.html

Lagoze, C. V. (2008). *ORE User Guide - Primer.* Retrieved from http://www.openarchives.org/ore/1.0/primer.

Lagoze, C. V. (2008, October). *ORE User Guide - Resource Map Implementation in Atom.* Retrieved from http://www.openarchives.org/ore/1.0/atom

McBride, B. (2004, February 10). *RDF Primer.* Retrieved from http://www.w3.org/TR/rdf-primer/

*Microsoft Research OreChem Project*. Retrieved from http://research.microsoft.com/en-us/projects/orechem/

*MPQC* . Retrieved from http://www.mpqc.org/

Peter Murray Rust, e. a. . *Semantic Comp Chem*. Retrieved from Semantic Comp Chem: http://bitbucket.org/gigadot/semantic-compchem/

*PubSubHubbub*. Retrieved from http://code.google.com/p/pubsubhubbub/

*Saxon HE Documentation.* Retrieved from http://www.saxonica.com/documentation/using-xsl/commandline.html

Saxon HE Download. Retrieved from http://saxon.sourceforge.net/#F9.2HE

*Triplestore wiki*. Retrieved from http://en.wikipedia.org/wiki/Triplestore

Young, D. Retrieved from The Absolute Beginners Guide to Gaussian: http://www.ccl.net/cca/documents/dyoung/topics-orig/gaussian.html