CCLRC
Daresbury Laboratory

Data management of CML marked up simulation data - AgentX

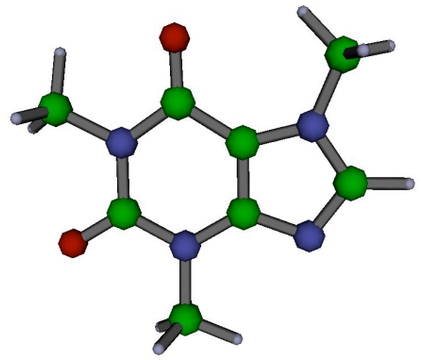
Philip Couch (CCLRC)
Toby White (Cambridge)

DL_POLY workshop, 04/05/06

Philip Couch
e-Science

CCLRC Daresbury Laboratory

eCCP Project

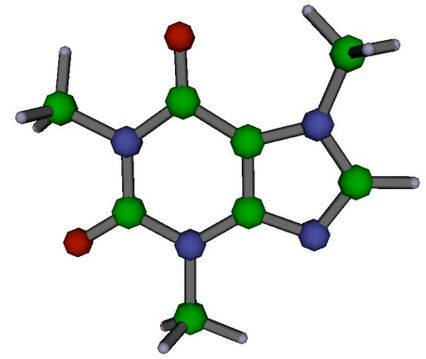


- e-CCP
  - e-Science for computational communities
  - Focus on information management
  - Application interoperability
  - W3C and semantic web standards and tools

- What are the CCPs
  - The UKs Collaborative Computational Projects
  - Cover a broad range of science (12 groups)
  - Develop, maintain and distribute computational codes
  - Promote the best computational methods (training and workshops)
  - Involve about 240 academic groups in most UK universities, and around 150 collaborating groups in the rest of Europe, the USA and Japan

Philip Couch
e-Science

Information extraction

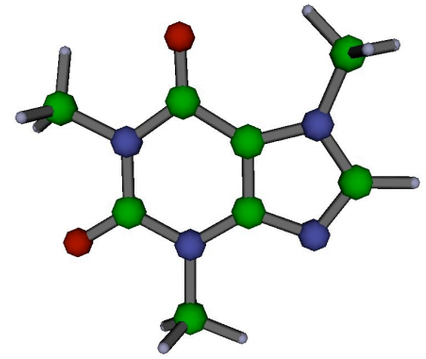


- How do we extract information from XML files?
- XML is a mature and well adopted standard
  - Many tools exist for parsing XML documents
- Standards exist for addressing parts of an XML document
  - XPath
  - Example: the x-coordinate of the first atom of the first molecule can be found using the XPath expression:

    ‘/molecule[1]/atom[1]/@x’

```
<molecule>
<atom x=“1.2” y=“0.2” z=“4.5” elementType=“C” />
<atom x=“2.1” y=“0.2” z=“3.1” elementType=“C” />
</molecule>
```

Problem



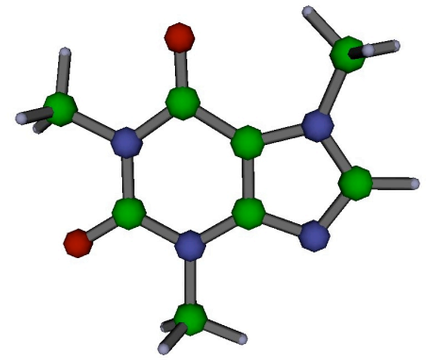
- Simulation codes can use existing XML tools to find data using XPath
  - But…

```
<molecule>
<atom x3="1.2" y3="0.2" z3="4.5" elementType="C" />
<atom x3="2.1" y3="0.2" z3="3.1" elementType="C" />
</molecule>
```

  - XPath = '/molecule[1]/atom[1]/@x' X
  - The x-coordinates are in a different place in the document!
  - You need to understand the structure of the document in detail (understand the data model) in order to find data using XPath

CCLRC Daresbury Laboratory

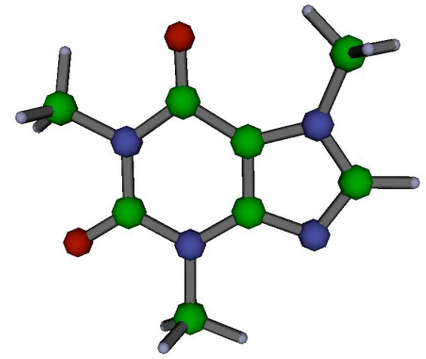
Problem



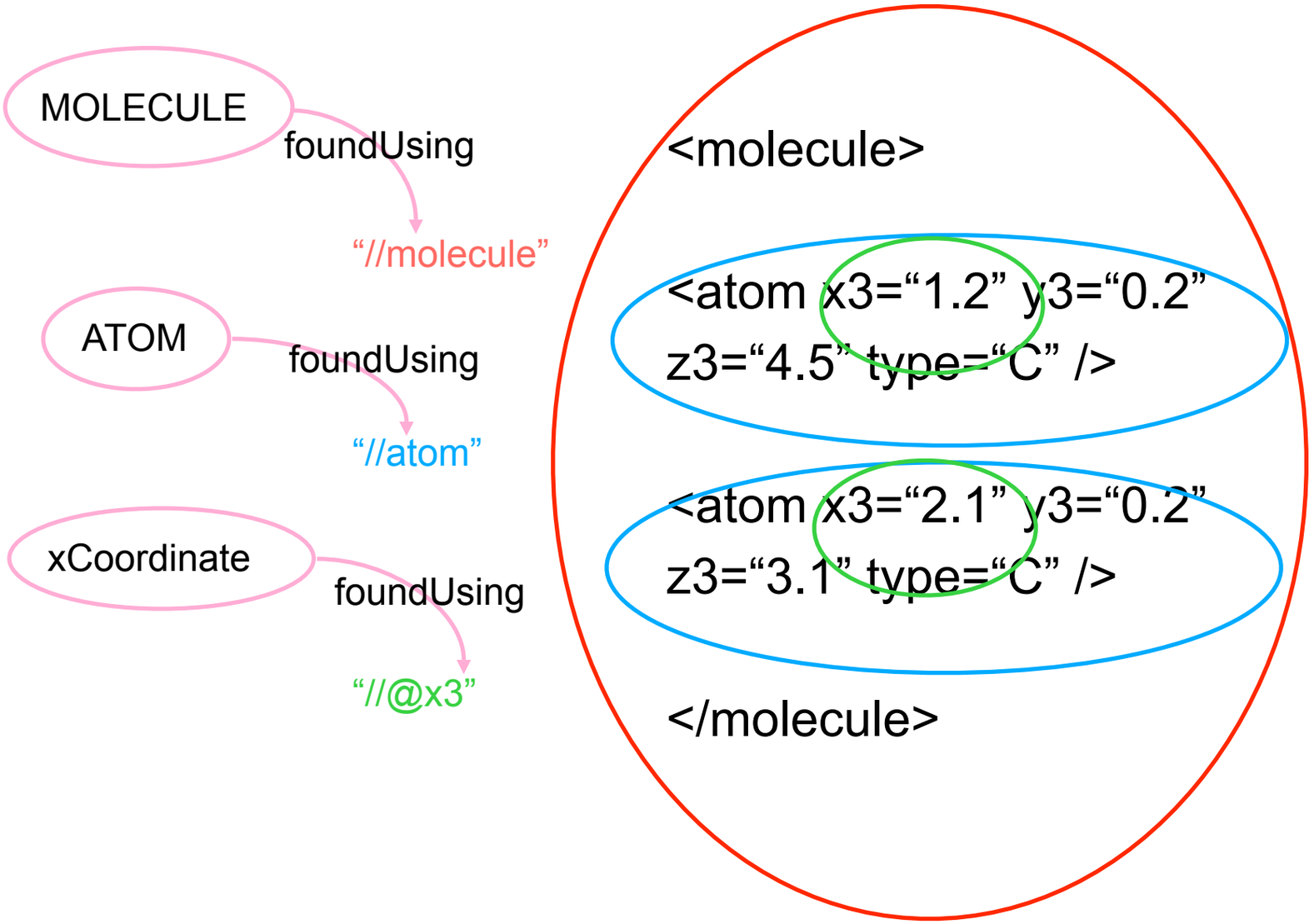
- Using XPath also requires an understanding of:
  - The XPath standard
  - The XML standard
  - The API of the XML parser
  - Wrappers for accessing the XML parser (particularly from Fortran)
    - FoX provides a promising solution (ask Toby White)

- Can we produce a tool that can be used to manage XML data that also:
  - Requires little understanding of data standards
  - Has a simple API
  - Can work with different data formats
  - Is easy to integrate with existing simulation and analysis tools

AgentX Framework - Overview



Terms Mappings Data



Specify how to locate data in a document that has a certain meaning

First, specify a set of terms that relate to things of interest

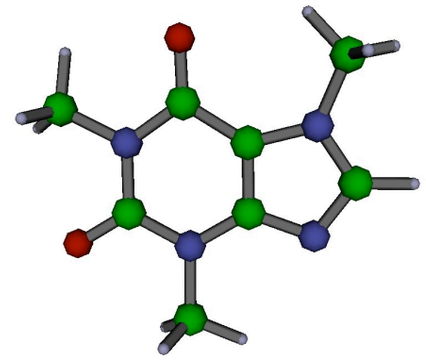
Associate these terms with XPath expressions

(terms are a shorthand for the XPath expression)

These terms can be used to locate parts of the document of interest

Philip Couch
e-Science

AgentX Framework - Ontology



Four components to the AgentX framework:

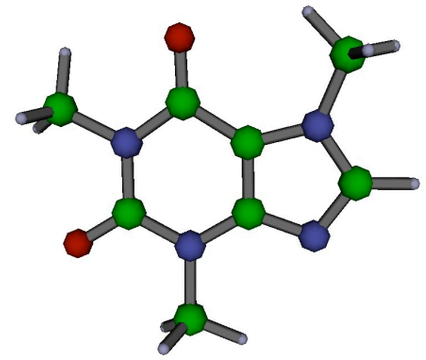
- Ontology:
  - Specifies standard terms (e.g. *Molecule*, *Atom*, *LatticeVector*)
  - Provides the meaning of the terms (e.g. *xCoordinate* is a property)
  - Adds restrictions
    - Example: cardinality (*Atom* has a single *xCoordinate*)
  - Can express class and property hierarchies

- The Ontology is represented using XML

- Can use existing standards (Web Ontology Language - OWL)

Philip Couch
e-Science

AgentX Framework – Mappings



- Mappings
  - Relates terms in the Ontology to data in documents using RDF (Resource Description Framework)

    'Data relating to *Atom* can be found by evaluating the XPath expression: // atom'

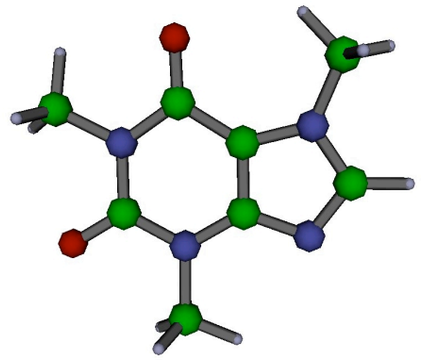
  - AgentX has various ways to identify parts of data documents

    'Data relating to *xCoordinate* are represented by the first sequence of non-white space characters of xyz3 attribute values'

  - Mappings are represented using XML

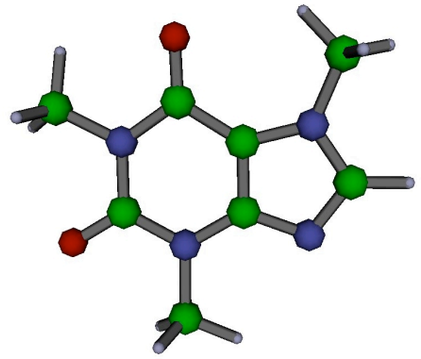
  - Each data format is associated with its own set of mappings
    - The XPath expression associated with a specific term may be different for different data formats
    - A user of AgentX works with the terms, not the XPath expressions
      - Lets the user work transparently with different data formats

AgentX Framework – Data



- Data
  - Currently works with XML data
  - Uses existing XML languages
    - Mappings for part of the Chemical Markup Language (CML)
      - Can be used to locate:
        - » Structures (Cartesian coordinates)
        - » Parameters
        - » Properties
        - » Metadata
  - Develop extensions to other languages were they are required
    - International effort
      - http://www.datarepresentation.org

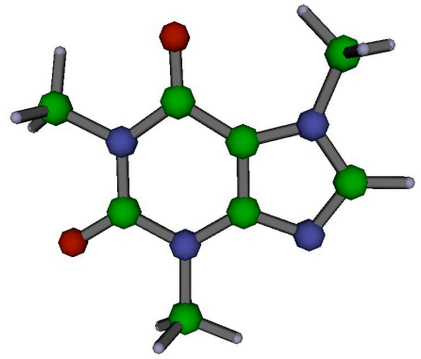
AgentX Framework - Tools

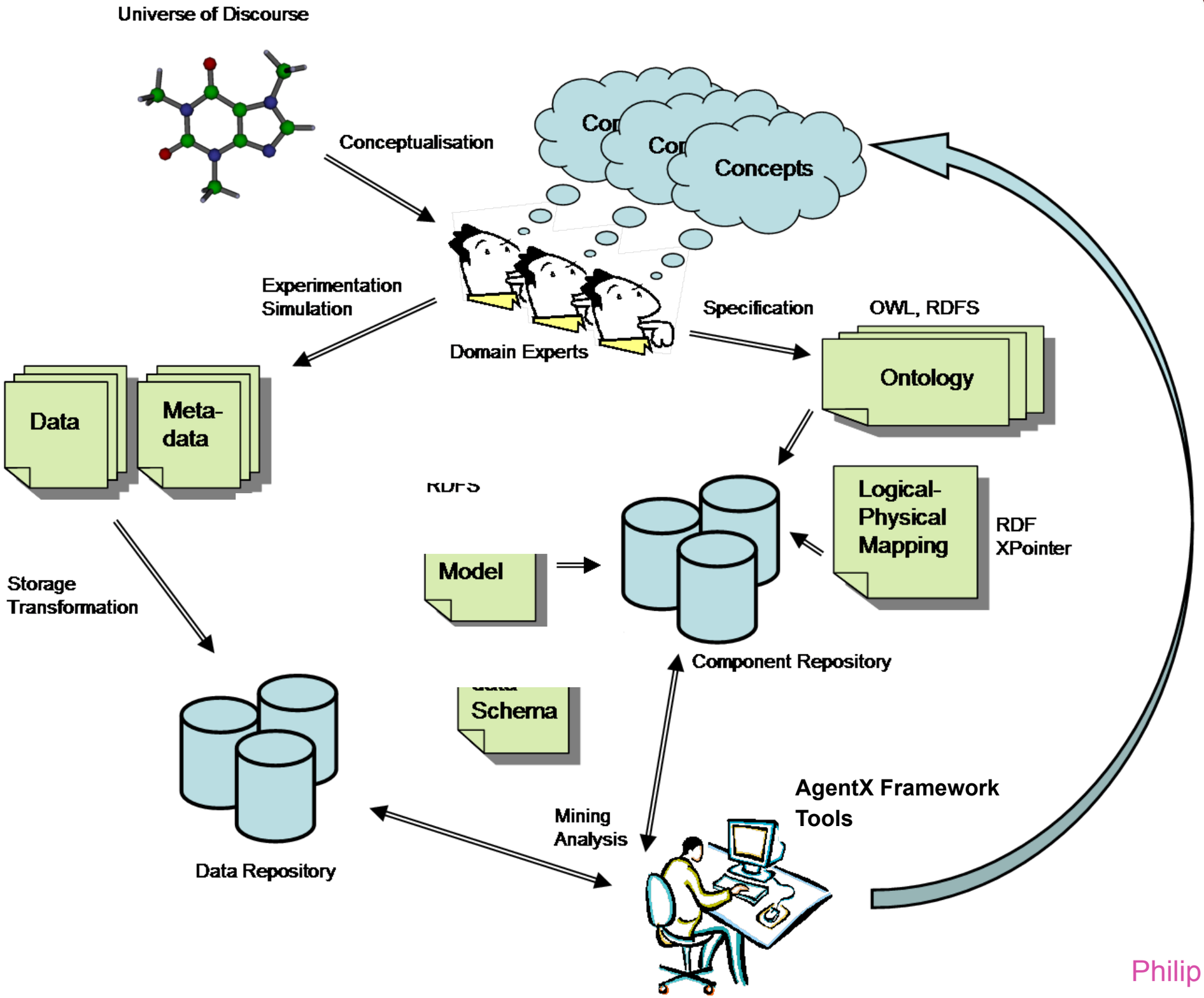


- Tools
  - AgentX core library (written in C)
  - Wrappers for Python, Perl and Fortran (77 and 90)
  - Simple API
    - Most information can be extracted using only 10 functions
  - Don't need to worry about the details of the XML (e.g. SAX, DOM, XML namespaces and navigating the document)
    - Queries are based on the terms in the Ontology

      'Find me the *xCoordinate* of the first *Atom* of the first *Molecule'*
    - AgentX does all the hard work!
  - You do **NOT** need to worry about the Ontology and mappings
    - Need to understand:
      - The terms used in the Ontology

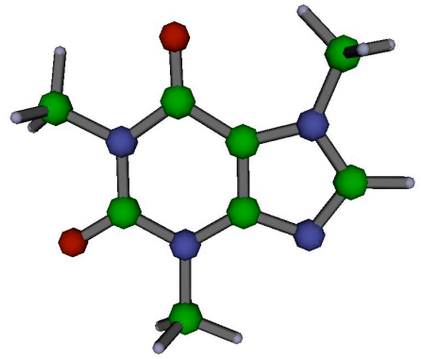
AgentX Framework – Illustrative

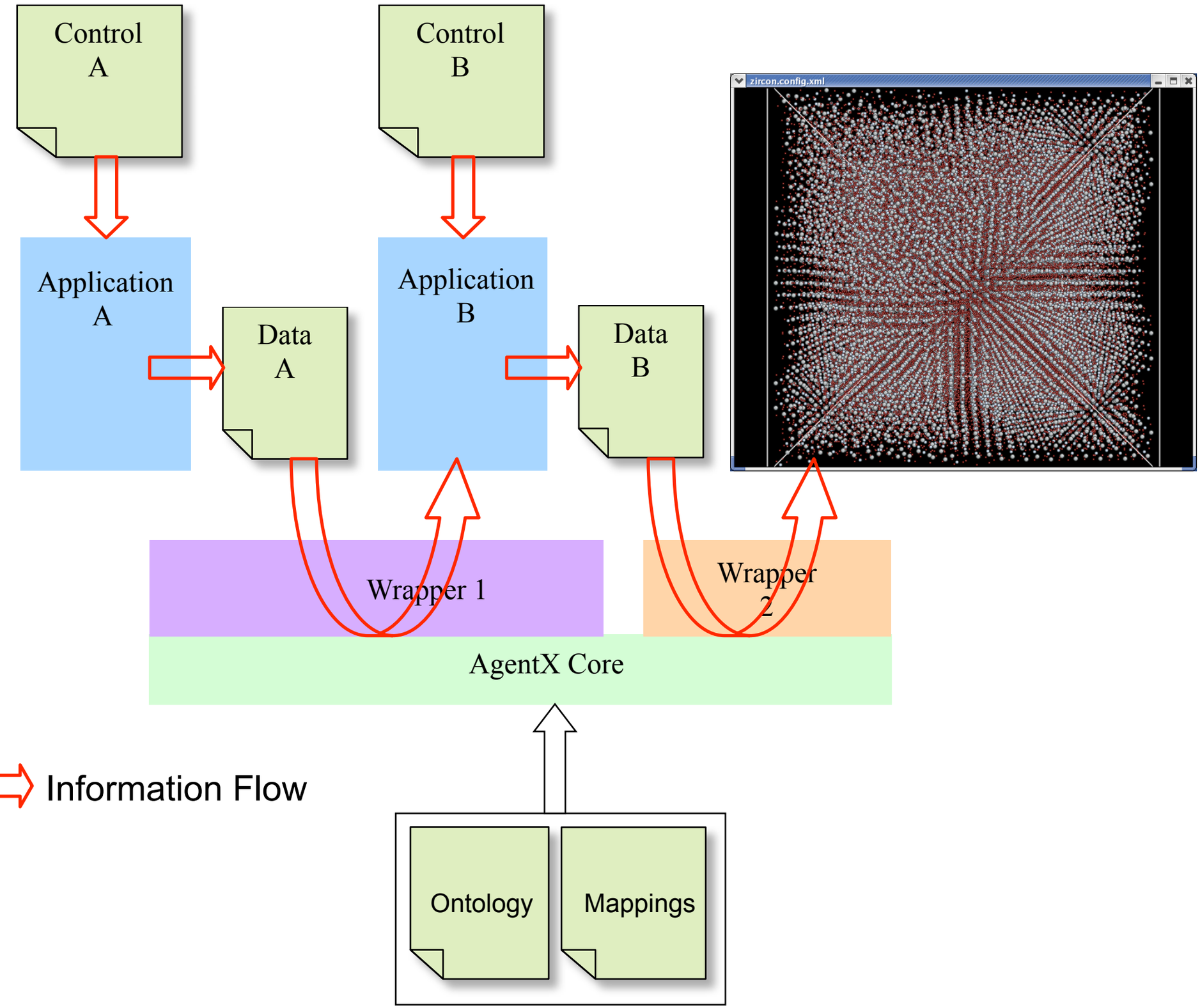




Philip Couch
e-Science

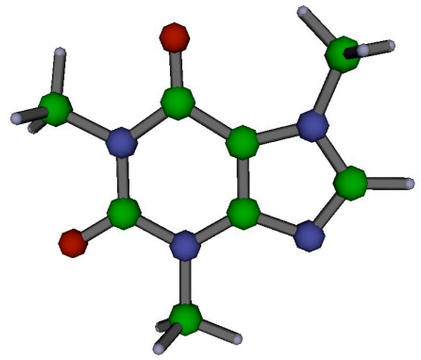
CCLRC
Daresbury Laboratory

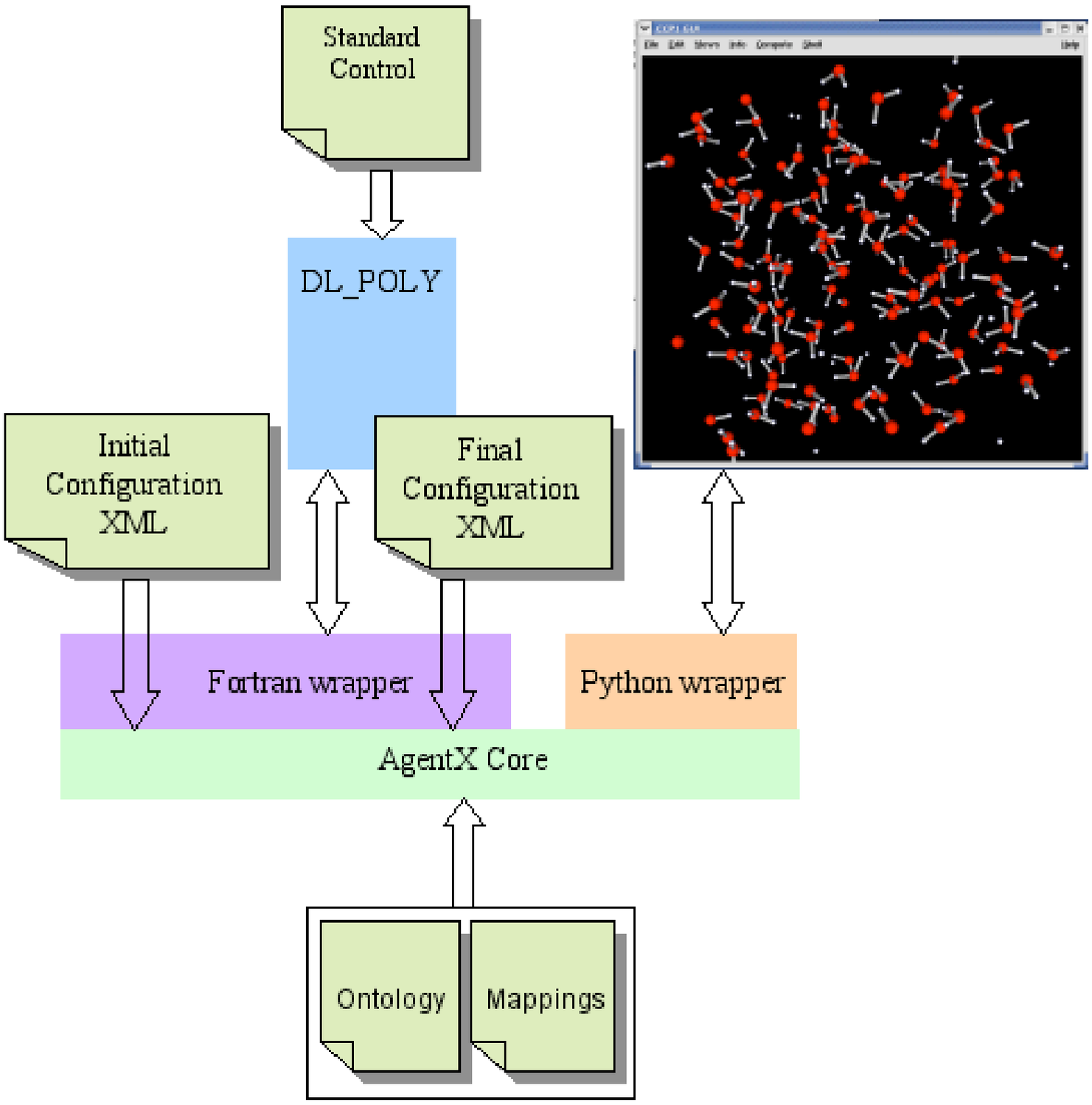




Philip Couch
e-Science

AgentX Framework – Demonstration



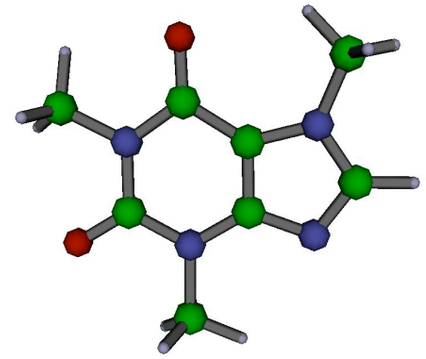


Launch demonstration 1

Launch demonstration 2

'hilip Couch
e-Science

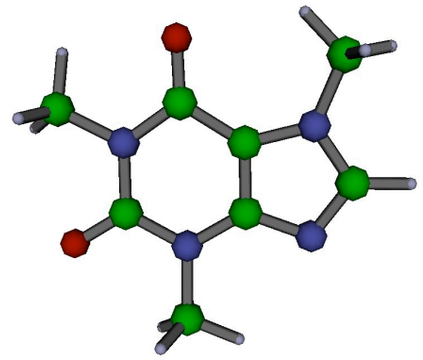
Fortran Example



```
retval=axParserStart()

retval=axDataGetUri("../xml/molpro.xml")

retval=axGetUri("../../ontology/ontology.owl")

retval=axGetUri("../../map/map.rdf")

retval=axBaseUri("http://www.grids.ac.uk/eccp/owl-ontologies#")

noMolecule=axSelect("Molecule")

do j=1,noMolecule

  noAtom=axSelect("Atom")

  do k=1,noAtom

    noprop=axSelect("elementType")
    call axValue(elementtype)
    retval=axDeselect()

    noprop=axSelect("xCoordinate")
    retval=axValueConvert(xcoordinate)
    retval=axDeselect()

    noprop=axSelect("yCoordinate")
    retval=axValueConvert(ycoordinate)
    retval=axDeselect()

    noprop=axSelect("zCoordinate")
    retval=axValueConvert(zcoordinate)
    retval=axDeselect()

    retval=axSelectNext()

  end do

  if (noAtom .gt. 0) then
    retval=axDeselect()
  endif

  retval=axSelectNext()

end do

if (noMolecule .gt. 0) then
  retval=axDeselect()
endif

retval=axParserFinish()
```

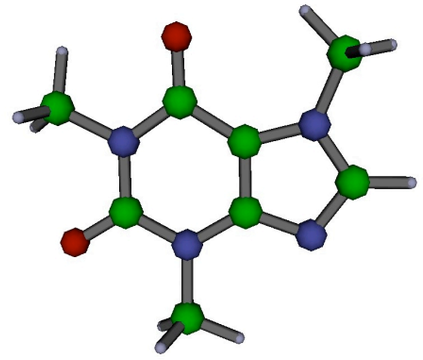
Philip Couch
e-Science

General Comments



- AgentX allows:
  - Efficient representation of data sets
    - 10^5 Atom data sets
  - Interoperability with different communities
    - Different mappings allow AgentX to work with a range of formats
      - VASP
      - Molpro
      - CML
      - DL_POLY
  - Straightforward integration with:
    - existing simulation codes
      - DL_POLY
      - SIESTA
      - GULP
      - GAMESS-UK
    - visualisation tools
      - AtomEye (Ju Li, MIT)
      - Ambrosia (UTOPIA, Manchester)
      - CCP1 GUI (CSE, DL)

Future Development



- Ontology and Mappings
  - Tools for visualising
    - What terms can be used for querying
  - Tools for creating/ extending
    - Can use existing tools to manage the Ontology
      - Protégé
- Adding support for non-XML data
  - AgentX is not tied to a specific syntax
  - NetCDF, HDF
- Further development to support very large data sets
  - (hybrid DOM/ SAX approach)
- Interfaces to databases
  - XML
  - Relational

Philip Couch
e-Science