ChemAxiom – An Ontological Framework for Chemistry in Science

Nico Adams,* Edward O. Cannon, Peter Murray-Rust
Unilever Centre for Molecular Science Informatics, University Chemical Laboratory,
University of Cambridge, Lensfield Road, Cambridge CB2 1EW, United Kingdom

Abstract

We present ChemAxiom as the first ontological framework for chemistry in science. ChemAxiom enables discourse about chemical objects in a computable language and is useful for the management of chemical concepts and data, the retrospective typing of resources, the identification of ambiguity and supports chemical text mining.

Ontology in Chemistry – The Current State of the Art

Chemistry is a central scientific discipline and at the heart of a number of other important sciences such as biomedical research. While the latter has derived tremendous benefit from the development of controlled vocabularies, taxomomies and ontologies for the annotation of biological knowledge and text, chemistry has been slow to adopt these technologies and remains, on the whole, an ontological wasteland, although Batchelor and others have made excellent cases for the use of (formal) ontological methods in chemistry.[1, 2]

There have been several attempts to apply ontological techniques to the field of chemistry in the past. Currently, the most widely used ontology in chemistry is the "Chemical Entities of Biological Interest" (ChEBI) ontology.[3] ChEBI contains ontological associations, which specify chemical relationships as well as the biological roles and applications of a molecule. A recent study by Batchelor showed, that ChEBI contains a substantial amount of implicit and disguised semantics, which significantly complicates its use in modern semantic information systems.[1] Other notable ontologies in the chemistry domain are the Chemical Ontology,[4] REX[5] and FIX,[6] which model physicochemical processes and methods respectively, as well as ChemTop, which is a subset of the BioTop ontology.[7] Though valuable for annotation, none of these efforts can be considered to constitute an ontological framework for chemistry.

The Case for Formal Ontological Methods in Chemistry

Chemical information systems and resource discovery in chemistry are often predicated on the use of chemical structure (connection table) as an identifier and as annotation for chemical data. This springs from the "central dogma" of chemistry, namely, that molecular structure is correlated to the physico-chemical and biological properties of chemical entities. While this practice has served a subsection of the chemical community relatively well, there are major problems: first and foremost, the use of a connection table as a chemical identifier leads to a fundamental ontological confusion between the universal "molecule" and a "real world" bulk substance. Yet, in many information systems, a physicochemical property of a *substance* is associated with the structure of a molecule. It does not make sense to speak of a melting point in terms of a molecule. Many physicochemical quantities are properties of the mereological sums of the molecules, which make up the substance and not properties of the molecules themselves. In practice, this almost always leads to "lossy" encoding of information and information compartmentalisation. Formal ontology can help by providing a clear distinction between the abstract notion of a molecule and a bulk substance as might be in use in the laboratory. A similar argument can be made for many identifiers: in many information systems, it is not clear whether the identifier applies to a molecule or the substance.

Many chemical entities have dynamic structures (*e.g.* rapidly interconverting isomers - glucose dissolved in water) and cannot be described by one structural representation alone, *i.e.* there exists a parthood relationship between a given chemical entity and the corresponding several structures that can be written. Furthermore, there is a dependence on the notion of time: the fluxional structure of a chemical entity is a function of time. Ontology can assist in defining and specifying both these parthood relationships as well as the time dependence.

Materials and formulations, too, can be composites of several molecular entities or other chemical entities, which, in turn can be composites. Moreover, the "history" (*e.g.* synthesis conditions, post-processing etc.) of a material often has a significant impact on its physical properties, which are not captured by simple structural annotation.

By adopting formal ontological methods, we can clarify ambiguous meanings: if, for example, text mining has identified the term "acid" in a piece of text, then it is not clear whether this refers to a molecule acting as an acid or a chemical substance (a bottle of acid). If, however, the term "pH" has also been identified in this context, a formal ontology could indicate that the concept "acid" refers to a substance rather than a molecule. When applied in this way, an ontology can be used to "retrospectively type" chemical objects - in this example into chemical substance or molecule.

The ChemAxiom Set of Ontologies

To address some of the points discussed above and to fill the ontological void that currently exists in the chemical domain, we have developed ChemAxiom - a set of separately maintainable, but interoperable and integrated ontologies (Figure 1).

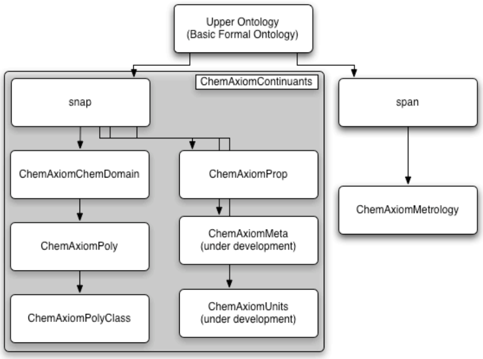

Figure 1: The ChemAxiom set of ontologies.

Each ontology describes a particular aspect of the chemical domain and collectively the ontologies form a framework for the description of chemistry. In designing ChemAxiom, we have borrowed from many bioscience ontologies such as the OBO family, ChEBI and MeSH and have derived some advantage from the fact that chemical concepts have clearer boundaries than biological ones. Consequently, ChemAxiom has been designed to (a) contain no implicit semantics, (b) be useful for the management of both chemical concepts and chemical data, (c) allow retrospective typing of chemical objects and the identification of ambiguity, (d) allow for undecideability either because of lack of knowledge or lack of axiomatisation and (e) allow for community extensibility and interoperability. Currently, the main use case for the ChemAxiom ontologies is the description of chemical data contained in documents of different types as well as machine output and the ability to support machine-generated RDF. We will present a formal evaluation of the ontologies w.r.t. this use case in further work. ChemAxiom complements other ontologies in the chemical field, which focus on, for example, compound taxonomy and biological function (ChEBI) or chemical structure (CO).[4] ChemAxiom has been prepared in the OWL language and is currently under active development, funded in part by both Unilever plc as well as the Dutch Polymer Institute. We are currently exploring the possibility of forming a broad platform around the ontologies with a number of partners and explicitly invite and value community participation in the development process. All ontologies are available at http://www.bitbucket.org/na303.

There are currently several ontology modules, which are integrated via the Basic Formal Ontology[8] as an upper ontology. ***ChemAxiomChemDomain*** is a small ontology which clarifies some fundamental concepts in chemistry, such as the parthood relationships between molecule and substance as well as generic roles which molecules and substances can assume. ***ChemAxiomProp*** currently contains a list of over 150 chemical and materials properties, together with definitions of symbols (where appropriate or available) and axioms for typing (see below). ***ChemAxiomMetrology*** is an ontology of over 200 measurement techniques and also contains a framework for instruments (though currently required metadata such as measurement conditions or specification of minimum information requirements are not included - this will be added at a later stage). It follows the same modeling pattern as *ChemAxiomProp* and thus also allows for typing of objects. ***ChemAxiomPoly*** and ***ChemAxiomPolyClass*** contain terms, which are in common use across polymer chemistry and materials science as well as a taxonomy of polymers in terms of generic chemical structure. ***ChemAxiomMeta*** will allow the specification of the provenance of chemical data and information. ***ChemAxiomContinuants***, finally, represents the integration of all of these sub-ontologies into an ontological framework for chemical continuants. Classes in all ontologies have natural language definitions (which have been omitted in the examples shown in this paper). Further ontologies will include ontologies of chemical reactions and experiments as well as specifying minimum information requirements for properties and measurement methods. We now illustrate some of the capabilities of the framework using a number of select examples.

Clarifying Parthood Relationships and Roles

Key concepts in the ChemAxiomChemDomain ontology are ChemicalIdentifier, ChemicalElement, MolecularEntity and ChemicalSpecies. We employ the IUPAC definitions of MolecularEntity and ChemicalSpecies and understand the former to be a "constitutionally or isotopically distinct atom, molecule, ion, […] etc., identifiable as a separately distinguishable entity", whereas a `ChemicalSpecies` is understood to be "an ensemble of chemically identical molecular entities". Following Batchelor's suggestion, we map `ChemicalElement, MolecularEntity` and `ChemicalSpecies` into the BFO as subclasses of `snap:Object`.[1]

```
ChemDomain:ChemicalSpecies
      a       owl:Class ;
      rdfs:subClassOf snap:Object ;
      rdfs:subClassOf
              [ a       owl:Restriction ;
                owl:onProperty
ChemAxiomProp:hasProperty ;
                owl:someValuesFrom
ChemAxiomProp:Property
              ] ;
      rdfs:subClassOf
              [ a       owl:Restriction ;
                owl:onProperty
ChemDomain:presentInAmount ;
                owl:someValuesFrom xsd:string
              ] ;
      rdfs:subClassOf
              [ a       owl:Class ;
                owl:unionOf ([ a
owl:Restriction ;
owl:onProperty ChemDomain:hasPart ;
owl:someValuesFrom ChemDomain:MolecularEntity
                           ] [ a
owl:Restriction ;
owl:onProperty ChemDomain:hasPart ;
                             owl:someValuesFrom
ChemDomain:ChemicalSpecies
                           ])
              ] ;
      owl:disjointWith ChemDomain:MolecularEntity ,
ChemDomain:ChemicalIdentifier ,
ChemDomain:ChemicalElement .
```

`ChemicalSpecies`, in turn, is composed of (`hasPart`) `MolecularEntity`(s) or other `ChemicalSpecies`. This crucial distinction now allows "real world" bulk substances (*e.g.* polymers, formulations, an amount of benzene in a bottle) to be modeled and kept ontologically distinct from the notion of the universal "molecule". Concepts such as `Solvent`, `Catalyst` or `Acid` are subclasses of either `ChemicalSpecies` or `MolecularEntity` as appropriate and are modeled in terms of roles: a `Solvent` is a `ChemicalSpecies` which has a role of `SolventRole`. While ChemAxiom makes parthood relationships specific, it is not easy to see how this can be reconciled with the current *de facto* use of many chemical identifiers, which are interchangeably applied to both molecules and substances (e.g. CAS numbers). If there is a unique molecular indentifier, such as InChI may be, then the identifier for the substance (URI) may be viewed as an aggregation of all the identifiers of the discrete molecular entities which are part of the substance. For materials, such as polymers, the situation is even more complex as it is difficult to discern a single uniqueness criterion: uniqueness in materials is often dependent on a material's history and context and it may be the case that several URIs may be required for the same material. This is an important question and subject to ongoing research.

Typing of Chemical Objects and Resources

ChemAxiomProp contains the central class `Property` (subclass of `snap:SpecificallyDependentContinuant`). `Property` has two types of subclass, `NamedProperty`, which is a primitive class and contains a list of concrete properties, which, too, are primitive. The other subclasses are mostly defined classes and represent categorizations in the domain. One `NamedProperty`, for example, is the `MeltingPoint`, which carries the following axiomatisation:

```
:MeltingPoint
      a       owl:Class ;
rdfs:subClassOf :NamedProperty ;
      rdfs:subClassOf
              [ a       owl:Restriction ;
                owl:onProperty :hasType ;
                owl:someValuesFrom
:ThermophysicalProperty
              ] ;
      rdfs:subClassOf
              [ a       owl:Restriction ;
                owl:hasValue "m.p."^^xsd:string ;
                owl:onProperty :hasSymbol
              ] .
```

In addition to being a direct `rdfs:subClassOf :NamedProperty`, `MeltingPoint` is a also a subclass of the anonymous class "hasType some ThermophysicalProperty" (l. 4-7). The defined class "`ThermophysicalProperty`", in turn, is modeled as the intersection of the two classes "`Property`" and "everything that is of type `ThermophysicalProperty`" (l. 5-12 below):

```
:ThermophysicalProperty
      a       owl:Class ;
      rdfs:label "Thermophysical properties"@en ;
      rdfs:subClassOf :Property ;
      owl:equivalentClass
              [ a       owl:Class ;
                owl:intersectionOf (:Property [ a
owl:Restriction ;
                           owl:onProperty :hasType
;
                           owl:someValuesFrom
:ThermophysicalProperties
                         ])
              ] .
```

Therefore, a reasoner will be able to infer that a `MeltingPoint` is also a subclass of `ThermophysicalProperty`. This is an example of both ontology normalization and retrospective typing; while all classes have asserted single inheritance, multiple inheritance can be inferred and maintained *via* a reasoner (ontology normalisation). Reasoning of this type can easily be accomplished using reasoners such as Pellet. Furthermore, we do not assert deep hierarchies - rather we allow a user to construct their own taxonomies using a combination of axioms and defined classes. If, for example, text-mining were to discover the term “melting point”, it could retrospectively be typed and therefore annotated as a `ThermophysicalProperty` or a `NamedProperty`.

Typing could be part of a larger system, in which a new “perspective” (*i.e.* a view onto a contextual reality, which need not be universally shared and may vary substantially and even conflict with other defined perspectives) can be constructed. This definition can be implemented *a posteriori* without needing to re-code the data. Typing of this sort is informing our object-oriented code generation in physical science applications. ChemAxiomProp does not yet contain notions of dimensionality, nor a subdivision of properties into qualities and dispositions. This is the subject of future development work.

Management of Chemical Data – Data in RDF

ChemAxiomContinuants is the result of the integration of all the sub ontologies discussed so far, and facilitates the modeling of chemical objects and data in RDF. We show how this can be done by creating an instance of the `NamedChemicalSpecies` benzene in ChemAxiomContinuants:

```
:benzene
      a       ChemDomain:NamedChemicalSpecies ;
      ChemAxiomProp:hasProperty
              :Density_1 , :BoilingPoint_1 ;
      ChemDomain:hasPart :benzeneMolecule .
:BoilingPoint_1
      a       ChemAxiomProp:BoilingPoint ;
      ChemAxiomProp:hasValue
              "80.1"^^xsd:string ;
      :measuredBy Metrology:Ebulliometry .
:Density_1
      a       ChemAxiomProp:Density ;
      ChemAxiomProp:hasValue
              "0.8786"^^xsd:string .
:benzeneMolecule
      a       ChemDomain:MolecularEntity ;
      ChemDomain:hasIdentifier
              :MolecularFormula_1 , :CASNumber_1 .
:CASNumber_1
      a       ChemDomain:CASNumber ;
      ChemDomain:hasValue "71-43-2"^^xsd:string .
:MolecularFormula_1
      a       ChemDomain:MolecularFormula ;
    ChemDomain:hasValue "C6H6"^^xsd:string .
```

In future work we will use the ontologies to describe data and chemical entities extracted from papers, theses and other sources of chemical information using our OSCAR3 entity extraction system. When coupled with the ability of retrospective typing of the extracted information, this opens the door to document classification and faceted search.[9]

Conclusions

The adoption of ontological methods in the chemistry domain is lagging far behind that of other disciplines. However, the integration of biomedical and chemical data is important for the future progress of science. We have developed a set of ontologies, that enables the description and typing of chemical objects and data in a semantically rich way. This work should go some way towards facilitating the integration of data from other scientific disciplines with chemical data.

References

1 Batchelor C (2008) An Upper Level Ontology for Chemistry. 5th International Conference on Formal Ontology in Information Systems:Saarbruecken, Germany

2 Frey JG, Hughes GV, Mills HR, et al. (2003) Less is more: lightweight ontologies and user interfaces for smart labs. UK e-Science All Hands Meeting:500-507 Nottingham, UK

3 de Matos P, Ennis M, Zbinden M, et al. (2006) ChEBI - Chemical entities of biological interest. http://www3.oup.co.uk/nar/database/summary/646, Accessed December 12, 2008

4 Feldman HJ, Dumontier M, Lng S, et al. (2005) CO: A chemical ontology for identification of functional groups and semantic comparison of small molecules. FEBS Letters 579:4685-4691

5 Degtyarenko K (2007) The Rex Ontology. http://obofoundry.org/cgi-bin/detail.cgi?id=rex, Accessed December 30, 2008

6 Degtyarenko K (2007) The FIX ontology. http://obofoundry.org/cgi-bin/detail.cgi?id=fix, Accessed December 30, 2008

7 ChemTop ChemTop. http://purl.org/chemtop/dev, Accessed February 28, 2009

8 Grenon P (2003) Spatio-temporality in Basic Formal Ontology. http://www.ifomis.org/Research/IFOMISReports/IFOMIS%20Report%2005_2003.pdf, Accessed Feb. 19, 2009

9 Corbett P and Murray-Rust P (2006) High-throughput identification of chemistry in life science texts. Computational Life Sciences II, Lecture Notes in Computer Science 4216:107-118